BorBann
A Real Estate Information Platform
(Project Proposal)

by

Pattadon Loyprasert 6510545608
Sirin Phungkun 6510545730

Submitted as
Software Requirements Specification
for
01219395 Innovative Software Group Project Preparation

Faculty of Engineering
Kasetsart University
Bangkok, Thailand

March 2025

BorBann
A Real Estate Information Platform
(Project Proposal)

by

Pattadon Loyprasert 6510545608
Sirin Phungkun 6510545730

This Project Submitted in Partial Fulfillment of the Requirements
for Bachelor Degree of Engineering
(Software Engineering)

Department of Computer Engineering, Faculty of Engineering
KASETSART UNIVERSITY
Academic Year 2025

Approved by:

Advisor ..... Date.../.../...
(Assoc.Prof.Dr. Kitsana Waiyamai)

Table of Contents

List of Tables

Chapter 1

Introduction

1.1 Background

The global real estate market shows positive growth trends despite various challenges. Real estate market remains a key sector for people like homebuyers and investors, influenced by complex characteristics with heterogeneous nature and affected by numerous elements like policies and consumer trends. The market is projected to grow from $4,143.71 billion in 2024 to $4,466.58 billion in 2025, and global real estate investment volumes continue to increase in 2024[1].

Thailand's real estate market faces both challenges and opportunities in the current economic climate. The real estate market in Thailand, especially in Bangkok in 2024-2025, faces challenges such as high household debt, strict credit conditions, and rising costs causing market contraction, decreasing project launches by 43.72% in 2024[2]. However, it also presents opportunities for recovery through foreign investment driven by tourism recovery, government incentives, and technological advancements. Condominiums remain attractive for rental yields and foreign buyers, while the housing market has moderate growth supported by infrastructure development.

Crucially, these market opportunities can significantly undermined by Thailand's real estate information ecosystem, which suffers from several structural limitations, including outdated listings, lack of official transaction data, and data fragmentation[3]. Addressing these issues is essential for maximizing Thailand's market potential and enabling effective decision-making for both homebuyers and investors seeking to capitalize on the available opportunities.

1.2 Problem Statement

Even with large amounts of real estate data available through various channels including property listings, historical transaction records, and news, investors still face challenges in aggregating, contextualizing, and deriving action from these sources. Additionally, the Thai real estate market has different characteristics from other countries because of the

limited availability of official property transaction records, listing duplication across platforms, and less standardized data compared to more mature markets.

Current platforms in Thailand like DDProperty, Hipflat, and Baania lack many important contextual factors such as climate risk assessments, neighborhood-specific news, and local beliefs that influence property valuation and real estate investment in the long term.

While platforms like Zillow and House Canary have advanced and comprehensive real estate analytics, predictive analytics, and user-friendly interfaces, they do not operate in Thailand. Even if these advanced platforms were to enter the Thai market, they would face challenges adapting their algorithms to the local context. Their models are calibrated to markets with standardized property classifications, valuations that vary by developers, and consistent transaction data—elements that are limited in Thailand's real estate ecosystem.

The BorBann platform addresses these challenges and aims to help users by creating a real estate data platform integrated with artificial intelligence, geospatial analytics, and data aggregation by focusing on analytics rather than transaction facilitation.

1.3 Solution Overview

BorBann will function as real estate data platform to users, integrating multiple data sources with advanced analytics. Below are the features and their details.

Customizable Automated Data Integration Pipeline

- **Automated schema inference:** Analyze website structures to identify and extract key data elements
- **Field mapping:** Recognize equivalent fields across different sources (e.g., "price" vs "cost")
- **Integration framework:** Seamless connection with data export systems
- **Multi-source support:** Process data from websites, APIs, and uploaded files

Retrain Model with Data from Pipeline

- **Custom prediction model:** Create custom prediction models by combining their pipeline data with platform data sources

Local Contextual Analytics

- **Environmental risk assessment:** Evaluate flood risk, natural disaster vulnerability, and air quality
- **Facility proximity analysis:** Calculate accessibility to schools, hospitals, transit, and commercial centers

- **Neighborhood quality scoring:** Generate composite metrics for area evaluation

Explainable Price Prediction Model

- **Feature importance analysis:** Quantify and rank factors influencing property prices
- **Adjustable characteristics modeling:** Adjust property characteristics to visualize price impacts
- **Confidence intervals:** Provide lower/upper price bounds for realistic expectations
- **Factor categorization:** Group influences by type (property features, location, market trends)
- **Natural language explanations:** Generate readable summaries of price determinants
- **Visual breakdowns:** Display contribution percentages and relationship graphs

Geospatial Visualization

- **Heatmap generation:** Create density visualizations for environmental factors, pricing, and metrics
- **Geospatial analytics:** Calculate analytics for custom geographic areas

1.4 Target User

User Type	Description	Needs
Real Estate Investors	Individuals focused on maximizing long-term investment, including foreign investors	• Investment analysis • Supporting data for decision making
Homebuyers	First-time purchasers, residents looking to relocate within Thailand, and expats seeking housing	• Property comparisons • Neighborhood insights • Pricing guidance

Table 1.1: Target Users and Their Needs

1.5 Benefit

The BorBann platform will provide numerous benefits to the Thai real estate market. It improves market transparency by enhancing accessibility to market information. It also helps both homebuyers and investors reduce research time, achieve lower transaction risks, and discover overlooked investment opportunities. Additionally, the platform will effectively represent the unique characteristics of the Thai real estate market.

1.6 Terminology

Term	Definition
Local Analytics	Analysis focused on extremely specific geographic areas, such as neighborhoods or even individual streets, to provide highly relevant insights.
Price Prediction Model	An algorithm or statistical model that forecasts property values based on historical data, market trends, and various property characteristics.
Proximity Analysis	The study of spatial relationships between geographic features, typically to evaluate the distance between properties and amenities or services.
Geospatial Visualization	The graphical representation of data with a geographic or spatial component, often through maps and interactive displays.

Table 1.2: Terminology

Chapter 2

Literature Review and Related Work

This chapter presents a review of existing platforms in the domain of real estate analytics and information systems. This review includes both international and Thai platforms, their features, strengths, and limitations. The analysis establishes the current state of real estate information platforms and identifies opportunities for the BorBann platform to address unmet needs in the Thai market.

2.1 Competitor Analysis

Many real estate platforms primarily function as property listing aggregators rather than comprehensive information systems. Their features support transaction facilitation through showcasing available properties, while offering limited analytical tools for market understanding. However, platforms like House Canary represent exceptions, operating specifically as information systems that provide investors with data analytics and market insights to support evidence-based decision-making in real estate investments.

Feature	BorBann (Proposed)	DDProperty	Hipflat	House Canary	Zillow
Customizable Automated Data Integration Pipeline	**Yes**	No	No	No	No
Retrain Model with Data from Pipeline	**Yes**	No	No	No	No
Local Contextual Analytics	**Yes**	No	No	Yes (Not optimized for Thailand)	Yes (Not optimized for Thailand)
Explainable Price Prediction Model	**Yes**	No	No	No	No
Geospatial Visualization	**Yes**	Yes	Yes	Yes	Yes

Table 2.1: Feature Comparison: BorBann vs Other Platforms

Table 2.1 demonstrates BorBann's technical advantages in the real estate analytics market. While all platforms offer geospatial visualization, BorBann's implementation includes advanced analytics like climate assessment, matching international platforms but

surpassing local Thai competitors. BorBann's automated data integration pipeline collects analytics-ready data automatically, unlike Thai platforms that rely on user inputs. Also, user can use that data to create their custom models. For local contextual analytics, BorBann provides Thailand-optimized insights including weather patterns and population density, whereas DDProperty/Hipflat only show basic nearby facilities, and international platforms lack Thailand-specific optimization. Most distinctively, BorBann's price prediction model prioritizes explainability and interpretability, revealing the reasoning behind valuations rather than presenting opaque predictions like competing platforms.

Technical Aspect	DDProperty	Hipflat	House Canary	Zillow	BorBann (Proposed)
Data Sources	User-submitted, Proprietary	User-submitted, Proprietary	Proprietary	Multiple Sources, Proprietary	Open Data, User-submitted
ML Implementation	Basic Prediction	None	Black-box Models	Black-box Models	Explainable Models

Table 2.2: Comprehensive Technical Comparison

Table 2.2 highlights key technical differences between BorBann and competing platforms. For data sources, while competitors rely heavily on government or user-submitted data, BorBann uniquely leverages open data combined with APIs to build a more comprehensive dataset. Regarding machine learning, BorBann distinguishes itself by implementing explainable models, providing transparency in its predictions. This contrasts with DDProperty's basic prediction capabilities, Hipflat's complete lack of ML features, and the black-box approaches of House Canary and Zillow where prediction logic remains hidden from users.

Chapter 3

Requirement Analysis

3.1 Stakeholder Analysis

The stakeholder landscape for BorBann includes primary stakeholders who directly interact with the platform and secondary stakeholders who indirectly influence its adoption and effectiveness. Recognizing these groups' specific needs helps develop a platform that delivers value across the real estate ecosystem.

Stakeholder Category	Stakeholder Group	Key Interests	Primary Requirements
Primary	Real Estate Investors	Environment risk assessment	Advanced analytics, predictive modeling, risk scoring
Primary	Homebuyers	Affordability, area quality, lifestyle alignment, environmental factors	User-friendly interface, neighborhood insights, price comparisons, risk assessments
Secondary	Real Estate Agencies	Market positioning, client advisement, property valuation	Property data, reliable market insights

Table 3.1: BorBann Platform Stakeholder Summary

The primary stakeholders are direct users who rely on BorBann's capabilities for informed real estate decisions. Their requirements range from investment analytics to intuitive neighborhood quality assessments. Secondary stakeholders like real estate agencies can influence platform adoption and serve as valuable data partners.

Primary Stakeholders

Primary stakeholders are those who directly use or are significantly affected by the BorBann platform. They rely on its capabilities to make informed decisions about real estate investments, purchases, and market trends.

Real Estate Investors

Real estate investors use BorBann to identify profitable investment, assess risks, and make data-driven decisions.

Interests and Concerns:

- Maximizing return on investment
- Identifying growth areas
- Risk assessment
- Diversification across neighborhoods and property classes

Requirements:

- Advanced analytics tools that is customizable
- Predictive price modeling with easy to understand explaination
- Risk scores in each area based on multiple factors

Homebuyers

Homebuyers rely on BorBann to find properties that align with their budget, lifestyle, and long-term needs. The platform helps them assess affordability, neighborhood quality, and potential risks.

Interests and Concerns:

- Property affordability and financing options
- Area quality
- Location amenities and lifestyle alignment
- Environmental factors including flood and pollution risks
- Commute time to work, schools, and essential services
- School districts and educational quality

Requirements:

- Intuitive, user-friendly interface with minimal learning curve
- Comprehensive neighborhood insights including safety metrics
- Price comparison tools with historical context

- Property quality and developer reputation metrics
- Detailed flood risk and environmental quality assessment
- Transit accessibility maps with time-based visualizations
- Education quality indicators and school zone mapping

Secondary Stakeholders

Secondary stakeholders are indirectly impacted by the BorBann platform, influencing its adoption, data availability, and overall market reach.

Real Estate Agencies

Real estate agencies act as intermediaries between property buyers and sellers. They use BorBann to improve their advisory services, gain a competitive edge, and provide market insights to clients.

Interests and Concerns:

- Market positioning relative to competitors
- Client advisement based on reliable data
- Access to comprehensive property information
- Accurate property valuation to support transactions

Influence:

- Potential data partners and platform promoters
- Can significantly influence adoption rates among clients
- May provide valuable transaction data not available elsewhere

3.2 User Stories

User stories capture the essential needs and goals of the BorBann platform's target users from their perspective. These stories follow the standard format: "As a [user type], I want to [action/feature] so that [benefit/value]."

Table 3.2: User Stories with Acceptance Criteria

User Story	Acceptance Criteria
Customizable Automated Data Integration Pipeline	
As a non-technical user, I want to input a website URL and have the system automatically generate a scraping configuration, so I can collect data without coding skills	• System provides visual confirmation of detected data structure • Non-technical users can successfully create a working pipeline in under 5 minutes
As a user, I want to paste multiple URLs from the same website pattern and have the system recognize the common structure, so I can efficiently collect data from similar pages	• System identifies common patterns across multiple URLs from the same website • A single configuration works for all provided URLs from the same pattern
As a user, I want to upload data files (CSV, JSON) as alternative data sources to integrate with my scraped data	• System accepts uploads of CSV and JSON files up to 50MB • Automatic schema detection for uploaded files with 90% accuracy
As a user, I want to customize the output format and template via an intuitive UI	• User can select from at least 4 output formats (JSON, CSV, SQLite, YAML) • UI provides visual preview of output format before confirmation
As a user, I want to schedule my data pipeline to run automatically at specified intervals	• Interface allows setting schedule frequency (hourly, daily, weekly, monthly) • Timezone selection is available for scheduling

Continued on next page

Table 3.2 – Continued from previous page

User Story	Acceptance Criteria
As a user, I want a dashboard showing all my data pipelines with their status and last run time	• Dashboard displays all pipelines with status indicators • Last run time and next scheduled run are clearly shown
Local Contextual Analytics	
As a user, I can see all contextual data for specific areas to make informed decisions	• System displays at least 5 contextual data points for any selected area • Historical trends for environmental factors available for at least 12 months
As a user, I want to see flood risk assessments for properties with historical flooding data	• Flood risk presented on a 5-point scale with historical context • System shows flood history for the past 10 years when available
As a user, I want to see daily air quality metrics around a property with historical trends	• Air quality index displayed with daily updates • Historical air quality data presented in trend graphs
As a user, I want to see all schools within a custom radius, including distance, ratings, and types	• System displays all schools within user-defined radius • Each school listing includes distance, type, and quality rating

Continued on next page

Table 3.2 – Continued from previous page

User Story	Acceptance Criteria
As a user, I want to view healthcare facilities near a property with distance and service information	• Healthcare facilities categorized by type (hospital, clinic, etc.) • Distance and basic service information provided for each facility
Explainable Price Prediction Model	
As a user, I want to see how specific contextual factors influence the property's predicted price	• System displays top 5 factors influencing price with percentage contribution • Visual indicators show positive/negative impact of each factor
As a user, I want the model to be interpretable so I can understand factors affecting the price	• Plain language explanations accompany each prediction • Interactive elements allow exploration of factor relationships
As a user, I want a statement or reason to back the prediction so I can trust the system's valuation	• Each prediction includes at least 3 specific supporting statements • System indicates confidence level for each prediction
As a user, I want to see a predicted price range for any property I select	• System shows lower and upper bounds for predicted price
As a user, I want to understand how the model derives the result so I can explain the valuation to others	• System provides visual breakdown of prediction process
Geospatial Visualization	

Continued on next page

Table 3.2 – Continued from previous page

User Story	Acceptance Criteria
As a user, I can see property listings on the map and click them to view detailed information	• Property details popup appears within 1 second of clicking a marker • Popup contains at least 5 key property attributes
As a user, I can see sections of the same property group to identify properties from the same development	• Properties from same development visually grouped with distinct boundaries • Group name appears when hovering over grouped properties
As a user, I can see multiple map visualization types to analyze different environmental factors	• System offers at least 5 different visualization overlays • Users can toggle between visualizations without page reload
As a user, I want to pan and zoom on an interactive property map to explore different areas efficiently	• Map responds to standard pan/zoom gestures within 100ms • Property markers adjust density based on zoom level
As a user, I want to set a custom radius around a point on the map to analyze the surrounding area	• User can place and adjust analysis radius on any map location • Contextual analytics update in real-time as radius is moved
Retrain Model with Data from Pipeline	

Continued on next page

Table 3.2 – Continued from previous page

User Story	Acceptance Criteria
As a non-technical user, I want to select one of my existing data pipelines as a source for model training	• User can select any active pipeline as a data source through a simple dropdown • System validates data compatibility before starting training process
As a user, I want to select from recommended model types appropriate for my data	• System suggests optimal model types based on data characteristics • Each model type includes a simple explanation of its strengths and use cases
As a user, I want to start the model training process with a single click after selecting my data sources	• Training begins with a single action after configuration • System provides confirmation that training has started
As a user, I want to see how accurate my trained model is compared to platform default models	• Performance metrics displayed with comparative benchmark against standard models • Visualizations show improvement or differences in prediction accuracy
As a user, I want to activate my newly trained model with a single click to apply it across the platform	• Model activation changes system behavior immediately • Visual indicator shows which model is currently active

Continued on next page

Table 3.2 – Continued from previous page

User Story	Acceptance Criteria
As a user, I want to see a list of all models I've trained with performance metrics and creation dates	• Management interface displays all user models with key metadata • Models can be sorted and filtered by different attributes
As a user, I want to receive clear explanations if my pipeline data is unsuitable for training	• System provides specific feedback about data quality issues • Suggestions for data improvements are provided when problems are detected

3.3 Use Case Diagram

The Use Case Diagram for the BorBann platform, shown in Figure 3.1, illustrates the primary interactions between the system and its three main user types: Real Estate Investors, Homebuyers, and Property Developers. The diagram captures the core functionality of the platform and how different users interact with its features.

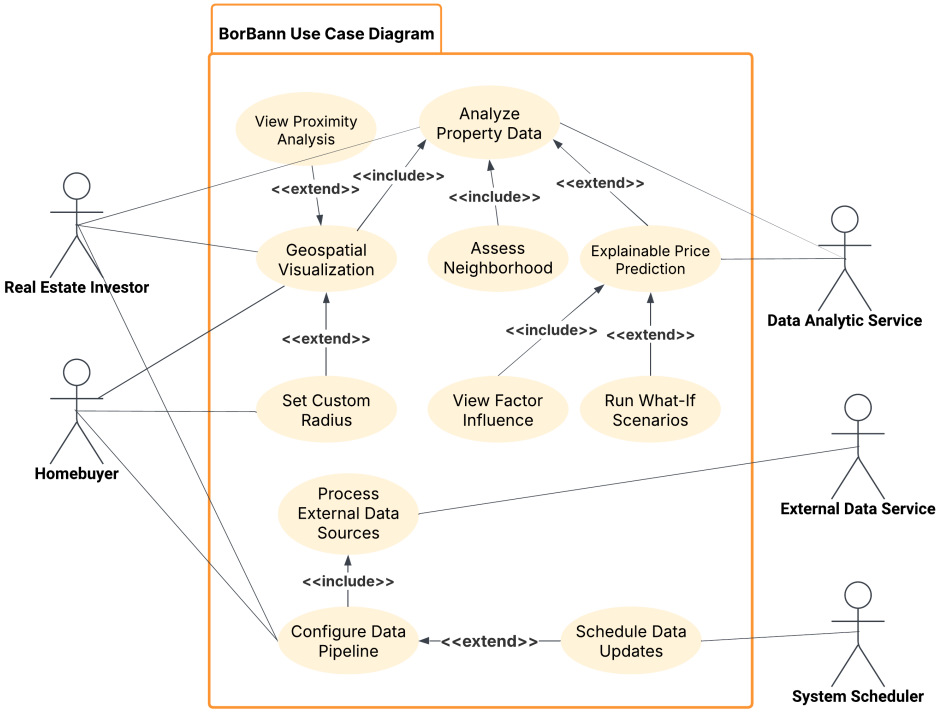

Figure 3.1: Use Case Diagram

Primary Actors

The use case diagram identifies five key actors interacting with the BorBann system:

- **Real Estate Investor** Seeks in-depth property analytics, market trends, and investment decision support.
- **Homebuyer** Interested in residential properties, lifestyle factors, and long-term value.
- **Data Analytic Service** Provides analytic capabilities to support prediction and data analysis features.
- **External Data Service** Supplies third-party data inputs via APIs and other integrations.

- **System Scheduler** Automates routine system tasks such as scheduled data refresh operations.

Core Use Cases

The BorBann platform provides the following primary functionalities:

- **Analyze Property Data** Allows users to examine detailed property information, metrics, and comparisons.
  - *Includes:* Assess Neighborhood
  - *Extended by:* Explainable Price Prediction
- **Geospatial Visualization** Interactive map-based visualization of property and neighborhood data.
  - *Extended by:* Set Custom Radius, View Proximity Analysis
- **Assess Neighborhood** Evaluates contextual factors around properties (e.g., schools, amenities).
  - *Includes:* View Factor Influence
  - *Extended by:* Check Environmental Risks
- **Explainable Price Prediction** AI-generated pricing insights with interpretable influencing factors.
  - *Includes:* View Factor Influence
- **Configure Data Pipeline** Enables setup of automated ingestion of external data from APIs and sources.
  - *Includes:* Process External Data Sources
  - *Extended by:* Schedule Data Updates

Extended Functionality

The following optional use cases extend the platform's functionality:

- **View Proximity Analysis** Triggered when users want detailed proximity-based data (e.g., nearby schools, transport).
- **Set Custom Radius** Adds user-defined range settings for map-based filtering and analysis.
- **Schedule Data Updates** Automates periodic refreshes of data pipelines using system scheduler logic.
- **Check Environmental Risks** Enables users to access data on environmental threats (e.g., flood zones, pollution).

Actor-System Interactions

Each actor interacts with the system in distinct ways, as shown in the diagram:

- **Real Estate Investor**
  - Initiates *Analyze Property Data*, *Geospatial Visualization*, and *Configure Data Pipeline*
  - Benefits from extended insights like *Set Custom Radius*, *View Proximity Analysis*, and *Explainable Price Prediction*
- **Homebuyer**
  - Accesses *Analyze Property Data*, *Geospatial Visualization*, *Assess Neighborhood*, and *Explainable Price Prediction*
  - Makes use of neighborhood-specific features like *View Factor Influence* and *Check Environmental Risks*
- **Data Analytic Service**
  - Collaborates with the system to enable *Explainable Price Prediction*
- **External Data Service**
  - Supports *Process External Data Sources* as part of the data ingestion workflow
- **System Scheduler**
  - Triggers *Schedule Data Updates* as an automated background process

Use Case Relationships

- **<<include>>** Represents mandatory sub-functions:
  - *View Factor Influence* is included in both *Assess Neighborhood* and *Explainable Price Prediction*
  - *Process External Data Sources* is included in *Configure Data Pipeline*
- **<<extend>>** Represents optional or conditional behavior:
  - *Set Custom Radius* and *View Proximity Analysis* extend *Geospatial Visualization*
  - *Check Environmental Risks* extends *Assess Neighborhood*
  - *Schedule Data Updates* extends *Configure Data Pipeline*

3.4 Use Case Model

The Use Case Model details the interactions depicted in the Use Case Diagram, describing each use case within BorBann's core features.

Use Case Name	**View Property Insights**
Actors	Real Estate Investor, Homebuyer, Property Developer
Description	Provides users with comprehensive analytics and contextual information about specific properties
Preconditions	User has selected a property from search results
Basic Flow	1. User selects a property for detailed viewing 2. System retrieves comprehensive property data 3. System presents property details with analytics 4. System displays neighborhood characteristics 5. System shows historical performance metrics 6. User reviews information
Alternative Flows	• User can extend to save the property as a favorite • User can extend to view explainable price predictions • User can request additional specific analytics
Postconditions	User gains comprehensive insights about the property
Associated Feature	Local Contextual Analytics - providing trend analysis and neighborhood-specific insights

Table 3.3: View Property Insights Use Case

Use Case Name	**Explainable Price Prediction**
Actors	Real Estate Investor, Homebuyer
Description	Provides transparent, interpretable price predictions with detailed explanations of contributing factors
Preconditions	User is viewing property insights
Basic Flow	1. User requests price prediction for a property 2. System calculates predicted price using its models 3. System generates explanation of factors influencing the prediction 4. System presents prediction with confidence interval 5. System displays factor weights and their impact 6. User reviews prediction and explanation
Alternative Flows	• User can adjust factors to see impact on prediction • User can compare prediction with market averages • User can save prediction for future reference
Postconditions	User understands the predicted price and the reasoning behind it
Associated Feature	Explainable Price Prediction Model - providing transparent explanations for price predictions

Table 3.4: Explainable Price Prediction Use Case

3.5 User Interface Design

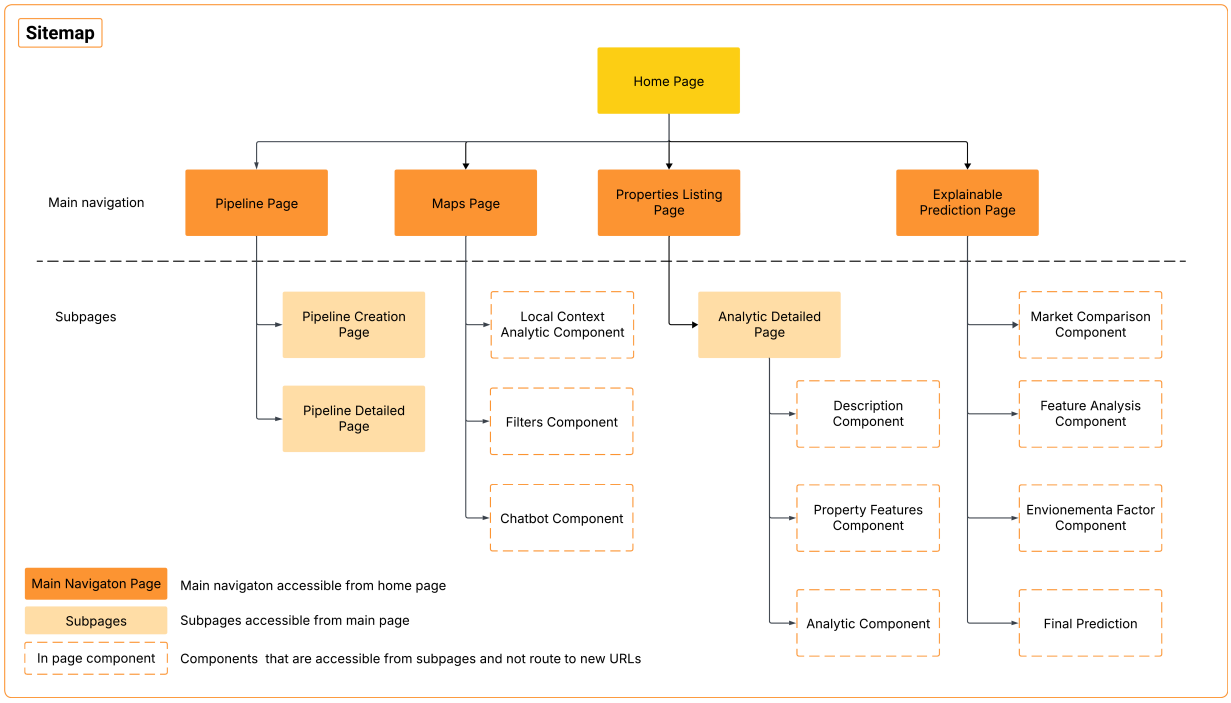

Figure 3.2: Sitemap - Platform Structure Overview

Figure 3.2 shows the structure of the BorBann platform. The Home Page acts as the central access point leading to four main sections: Pipeline, Maps, Properties Listing, and Explainable Prediction pages.

Each main section connects to specific subpages. The Pipeline section includes Creation and Detailed pages. The Maps section features Local Context Analytics, Filters, and Chatbot components. The Properties section provides Analytic Detailed pages with Description, Property Features, and Analytics components. The Explainable Prediction section offers Market Comparison, Feature Analysis, Environmental Factor analysis, and Final Prediction components.

This structure organizes the platform's key functions in a logical flow, making it easy for users to navigate between data management, visualization, and prediction features.

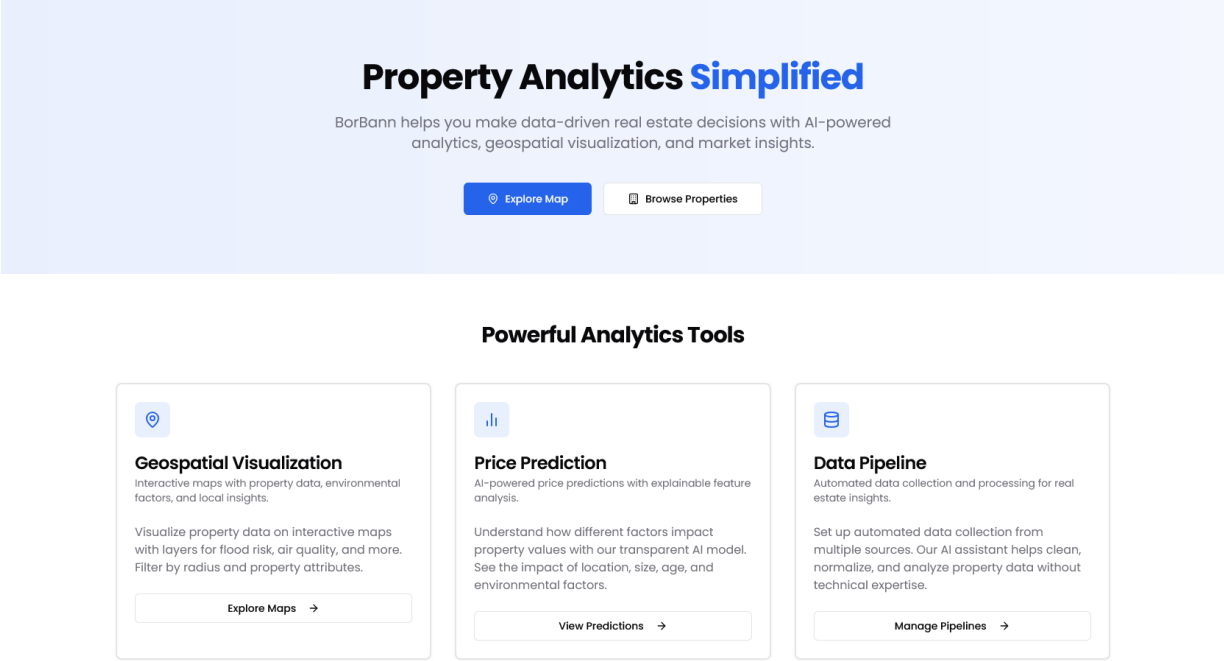

Figure 3.3: BorBann Platform Homepage

Figure 3.3 shows the BorBann platform homepage, which provides an intuitive entry point to the system. The page presents the core value proposition with "Property Analytics Simplified" and briefly explains how BorBann helps users make data-driven real estate decisions through analytics, geospatial visualization, and market insights. Two primary action buttons - "Explore Map" and "Browse Properties" - enable quick access to key functionality. The lower section showcases three main analytics tools: Geospatial Visualization for interactive property mapping, Price Prediction with explainable AI features, and Data Pipeline for automated data collection and processing.

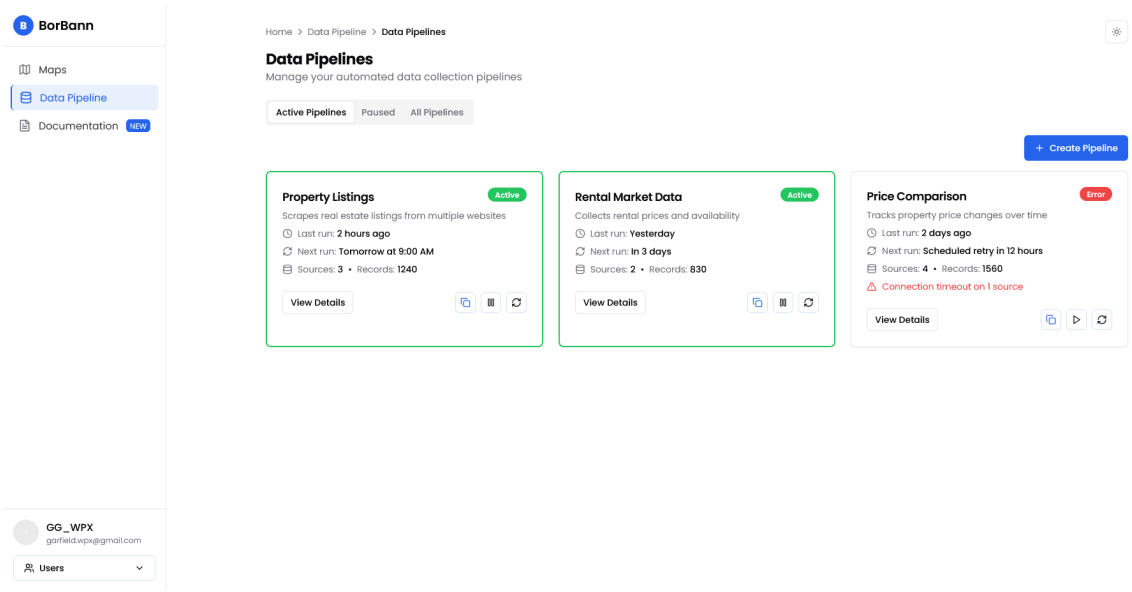

Figure 3.4: Data Pipeline Management Interface

Figure 3.4 showcases the Data Pipeline management dashboard, providing users with comprehensive control over their automated data collection processes. The interface features intuitive navigation with filtering options for Active, Paused, and All Pipelines, enabling efficient workflow management.

Each pipeline card presents essential operational metrics including last run timestamp, next scheduled execution, number of data sources, and total records processed. This at-a-glance view allows users to quickly assess data freshness and monitor collection performance across multiple pipelines.

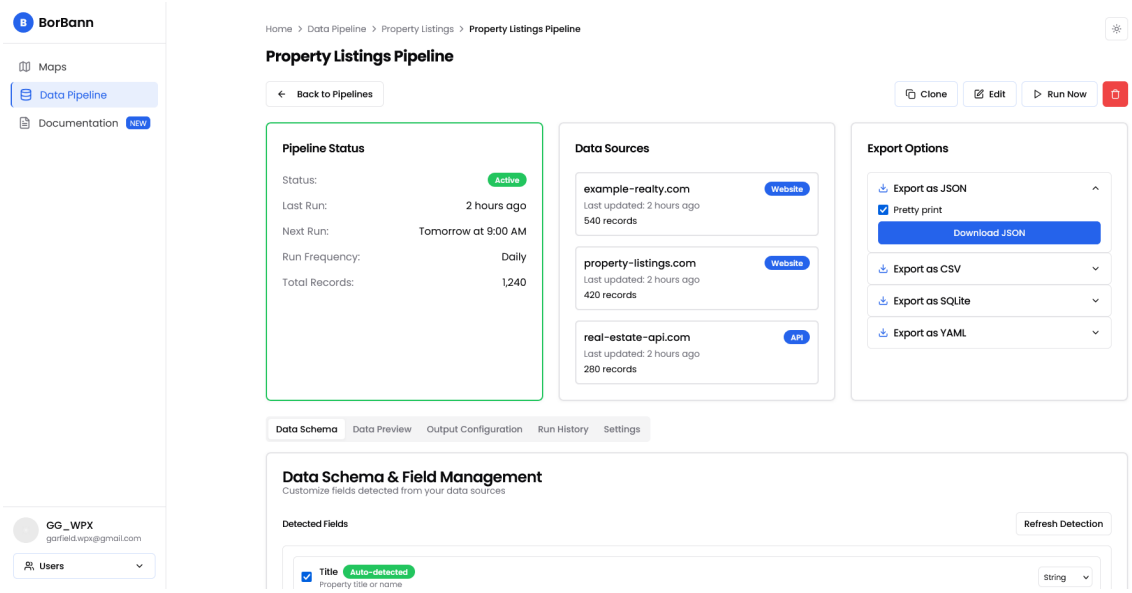

Figure 3.5: Data Integration Pipeline Detail - Overview

Figure 3.5 illustrates the pipeline detail view, highlighting key operational components including current status indicators, connected data sources, and available export formats. This overview provides users with immediate visibility into pipeline configuration and functionality.

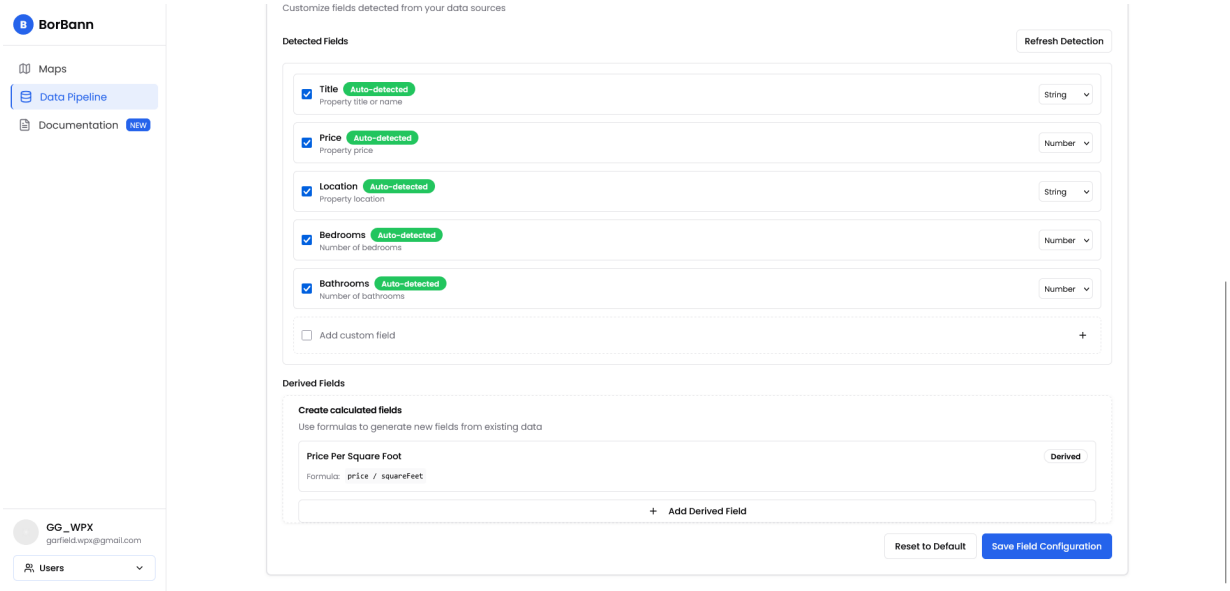

Figure 3.6: Data Integration Pipeline Detail - Field Management

Figure 3.6 displays the field management section of the pipeline detail page. This interface empowers users to customize their data structure by managing output fields and creating derived fields through a visual formula builder. Users can transform raw data into meaningful metrics without writing code.

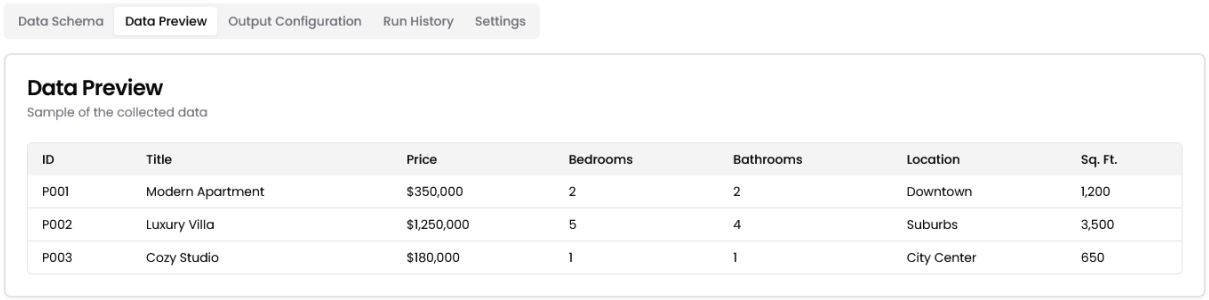

Figure 3.7: Data Integration Pipeline Detail - Output Data Preview

Figure 3.7 presents the data preview tab, where users can examine sample records generated by their pipeline. This real-time preview functionality allows users to validate data quality and structure before export or analysis.

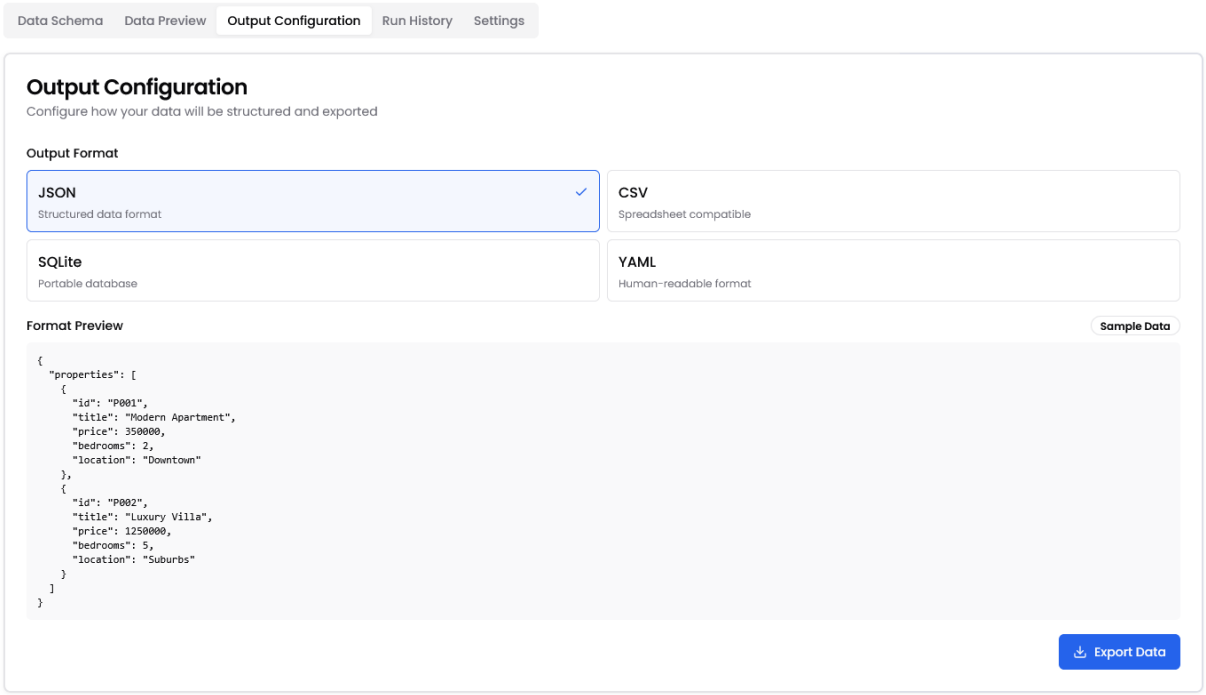

Figure 3.8: Data Integration Pipeline Detail - Export Configuration

Figure 3.8 shows the export configuration interface, where users can precisely define output schemas for their data exports. This tab enables users to select specific fields, customize formatting, and choose from multiple export formats to meet their downstream requirements.

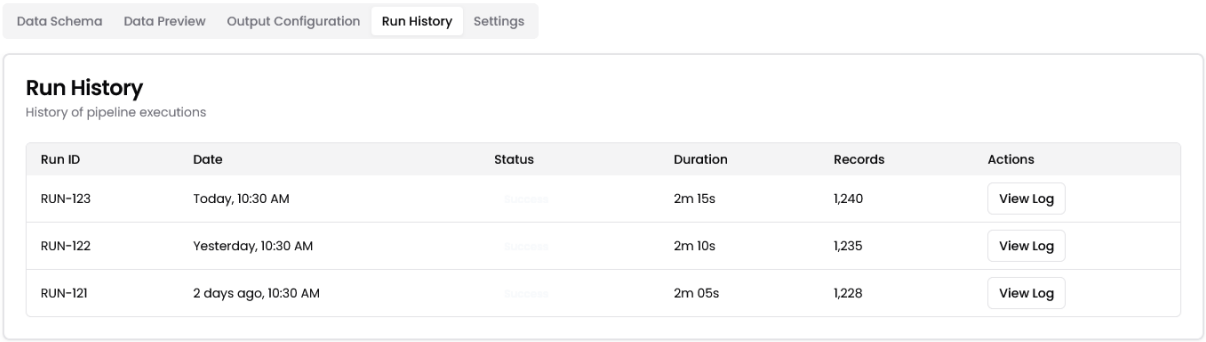

Figure 3.9: Data Integration Pipeline Detail - Run History

Figure 3.9 displays the run history tab, providing a chronological log of pipeline executions. This audit trail includes execution timestamps, duration metrics, and status indicators, enabling users to monitor performance trends and troubleshoot any execution issues.

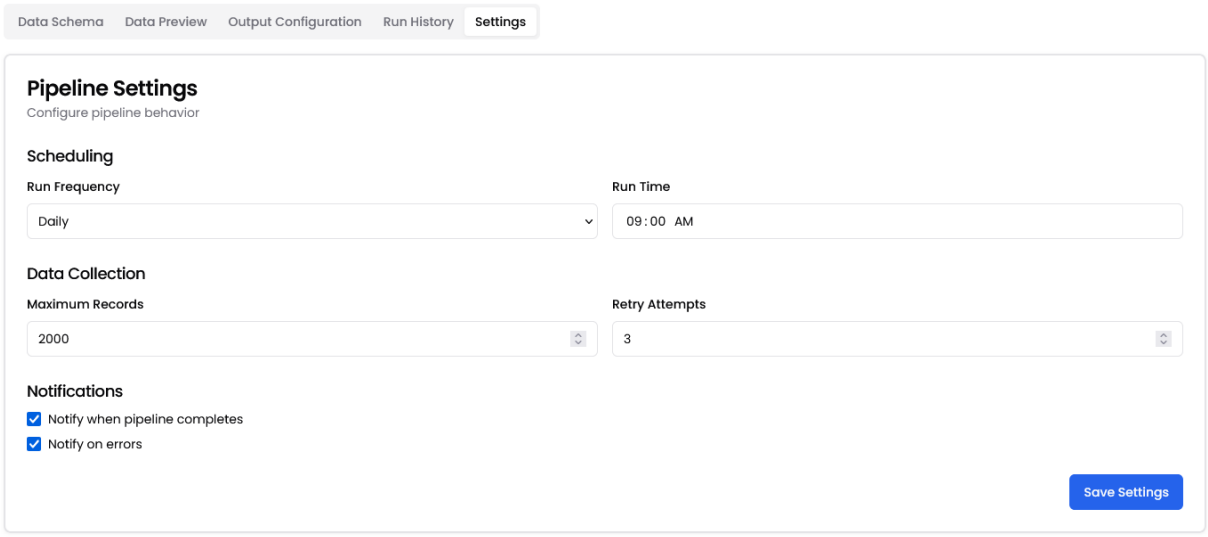

Figure 3.10: Data Integration Pipeline Detail - Pipeline Settings

Figure 3.10 illustrates the pipeline settings tab, where users can configure automation parameters including execution schedules, notification preferences, and retry policies. These controls allow users to establish reliable data collection routines that align with their operational requirements.

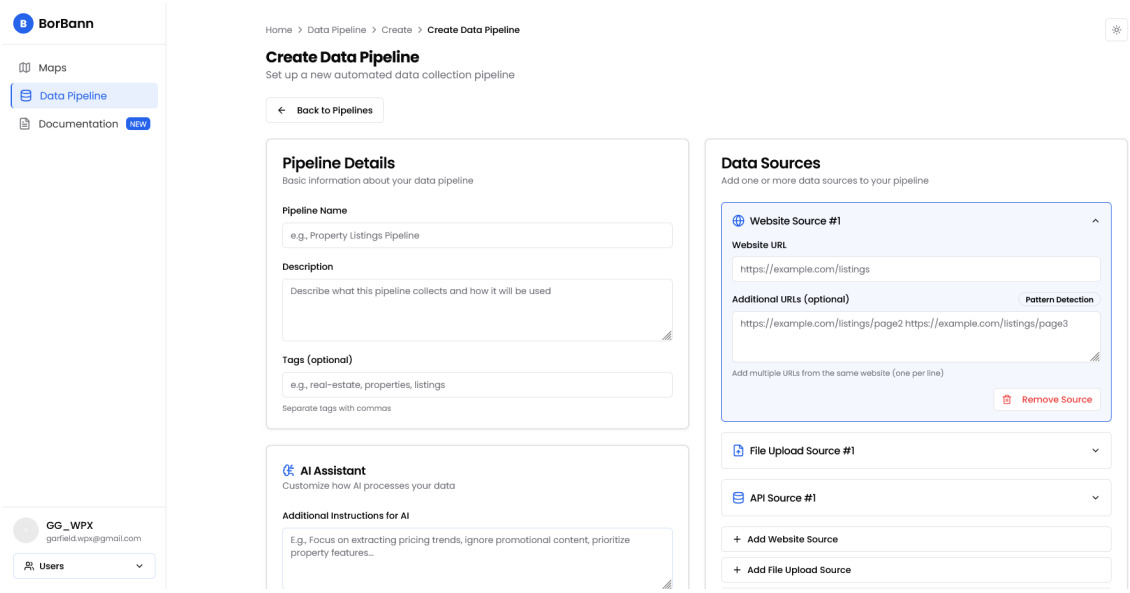

Figure 3.11: Data Integration Pipeline - Creation Interface

Figure 3.11 shows the pipeline creation interface where users input website URLs for automated data extraction. The form enables non-technical users to configure scraping operations without coding knowledge.

Figure 3.12 displays automation settings for pipeline execution with scheduling options and AI-assisted extraction configuration. Users can set run frequency and notification

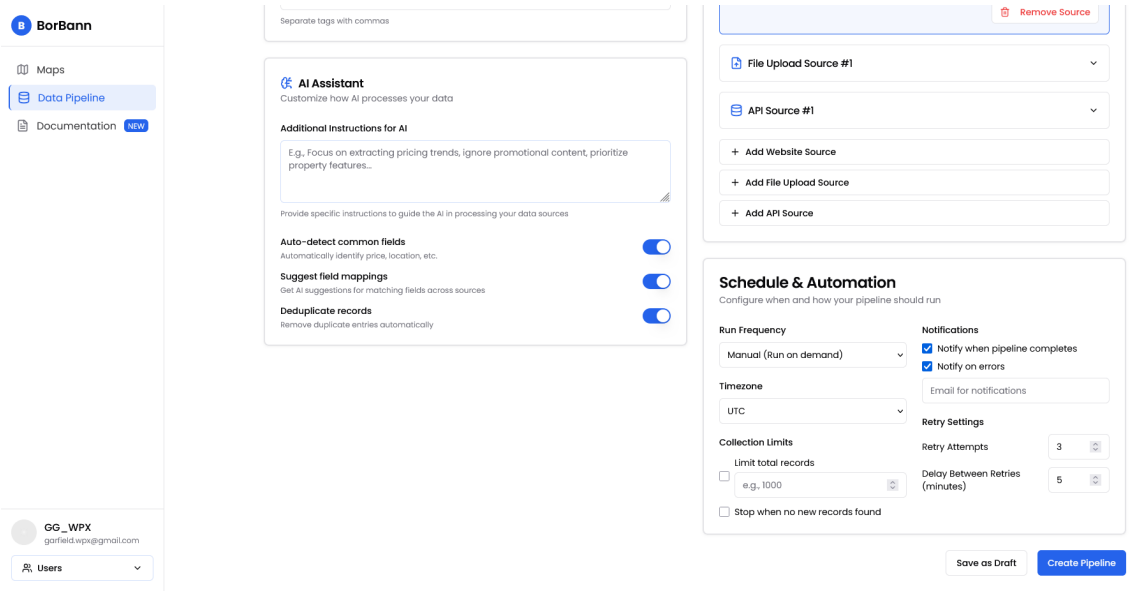

Figure 3.12: Data Integration Pipeline - Advanced Configuration and Scheduling

preferences to maintain automated data collection.

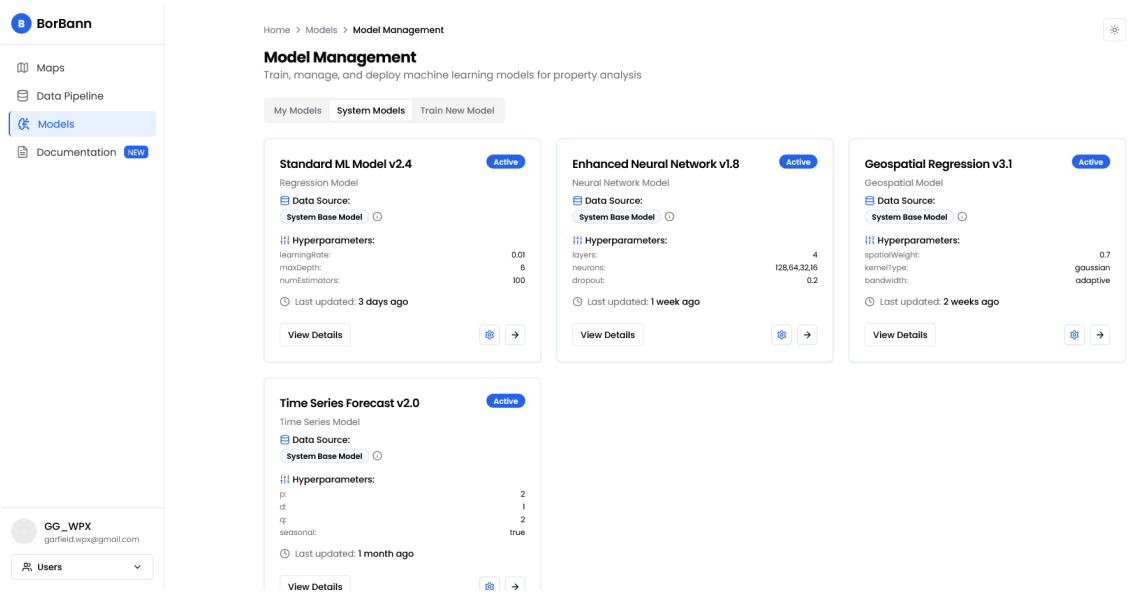

Figure 3.13: Model Management Dashboard

Figure 3.13 presents the model management dashboard where users can view and control prediction models. The interface displays performance metrics and allows single-click model activation.

Figure 3.14 shows the model creation interface for configuring new prediction models. Users can select data pipelines as training sources and choose from recommended algorithms with explanations of their strengths.

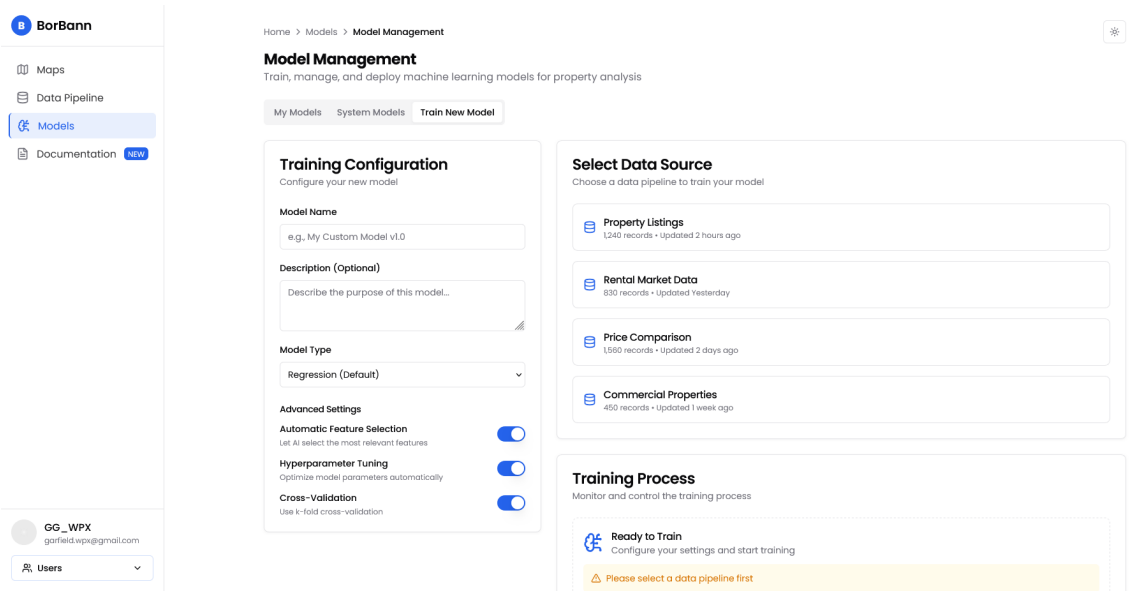

Figure 3.14: Model Creation Interface

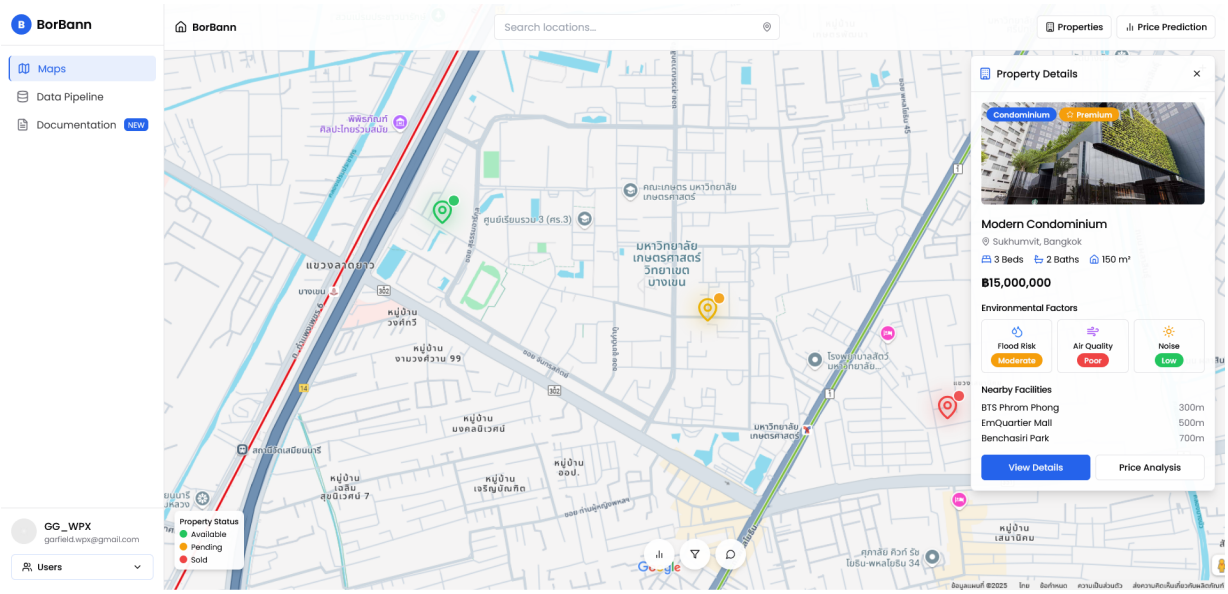

Figure 3.15: Geospatial Visualization - Main Map Interface

Figure 3.15 displays the interactive property map with color-coded markers and navigation controls. The interface supports pan/zoom gestures and provides filtering options through the sidebar.

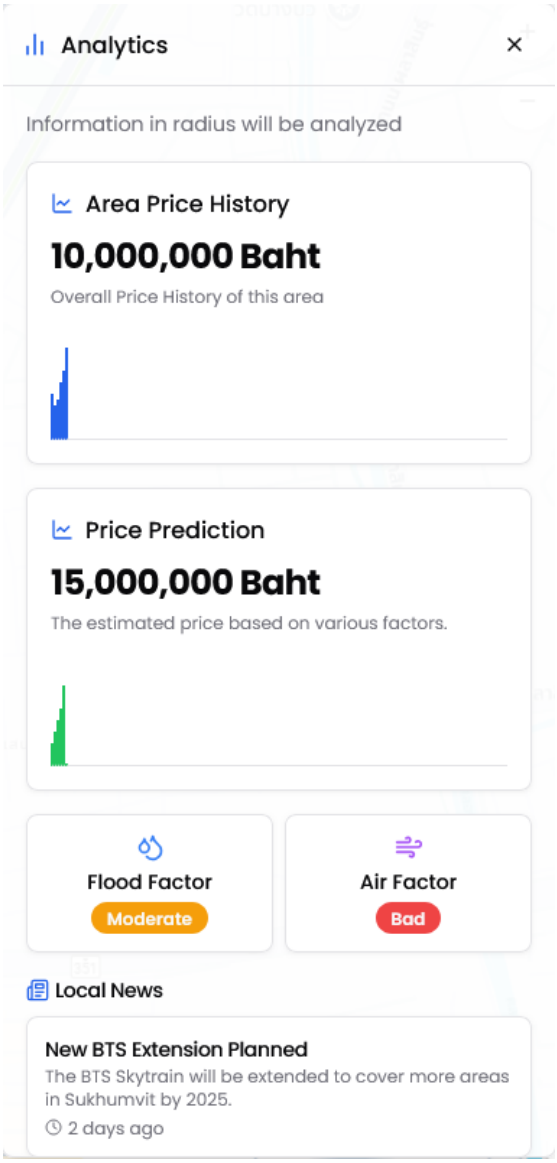

Figure 3.16: Map Analytics Overlay

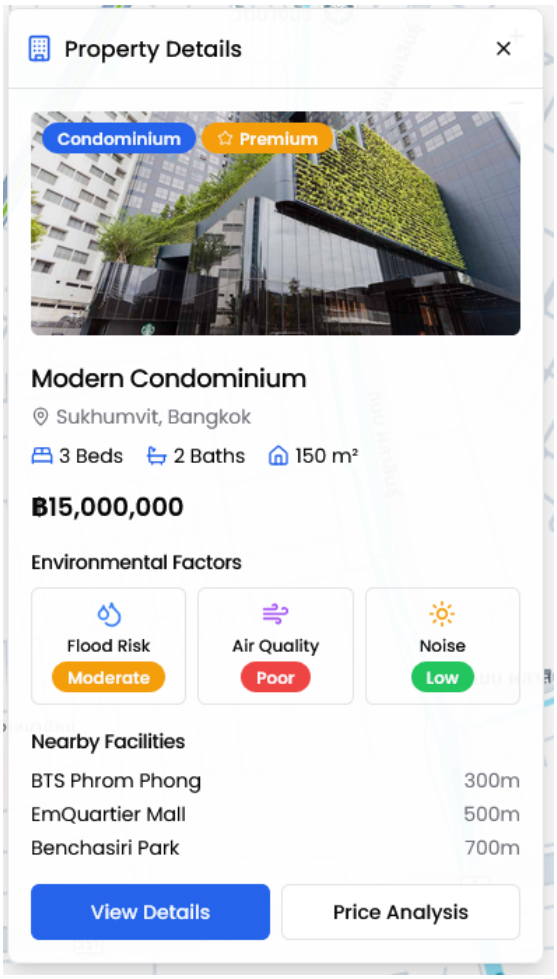

Figure 3.17: Property Detail Overlay

Figure 3.16 shows environmental factor visualization through heat maps and data layers. This overlay helps users analyze contextual factors like pollution levels directly on the map.

Figure 3.17 displays the popup that appears when users click map markers. This overlay provides immediate property information without requiring navigation away from the map.

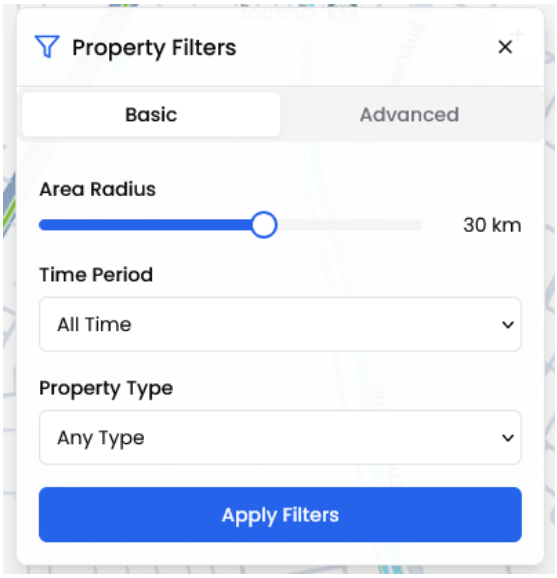

Figure 3.18: Property Filter Panel

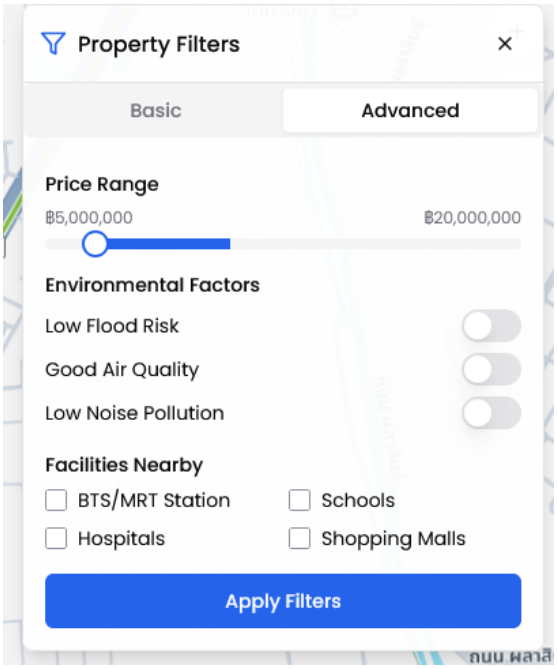

Figure 3.19: Advanced Property Filter Options

Figure 3.18 shows basic property filtering controls for price range, type and size parameters. The panel uses sliders and checkboxes for intuitive refinement of map results.

Figure 3.19 displays extended filtering options for specific amenities and neighborhood characteristics. These advanced parameters enable precise property matching based on detailed criteria.

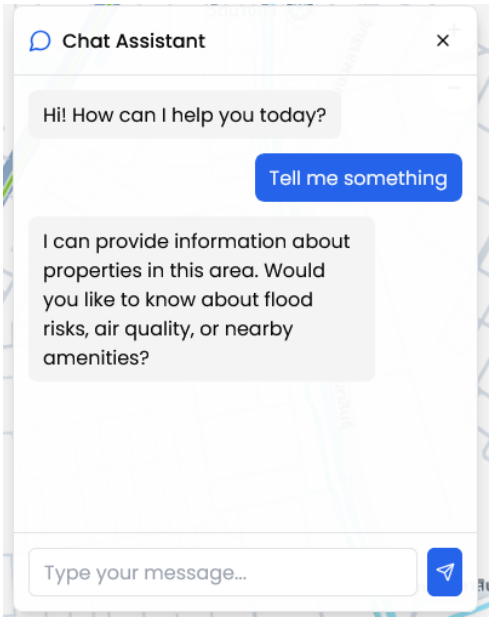

Figure 3.20: Interactive Chatbot Assistant

Figure 3.20 shows the conversational assistant for property search and analysis. The chatbot handles natural language queries about properties and market conditions.

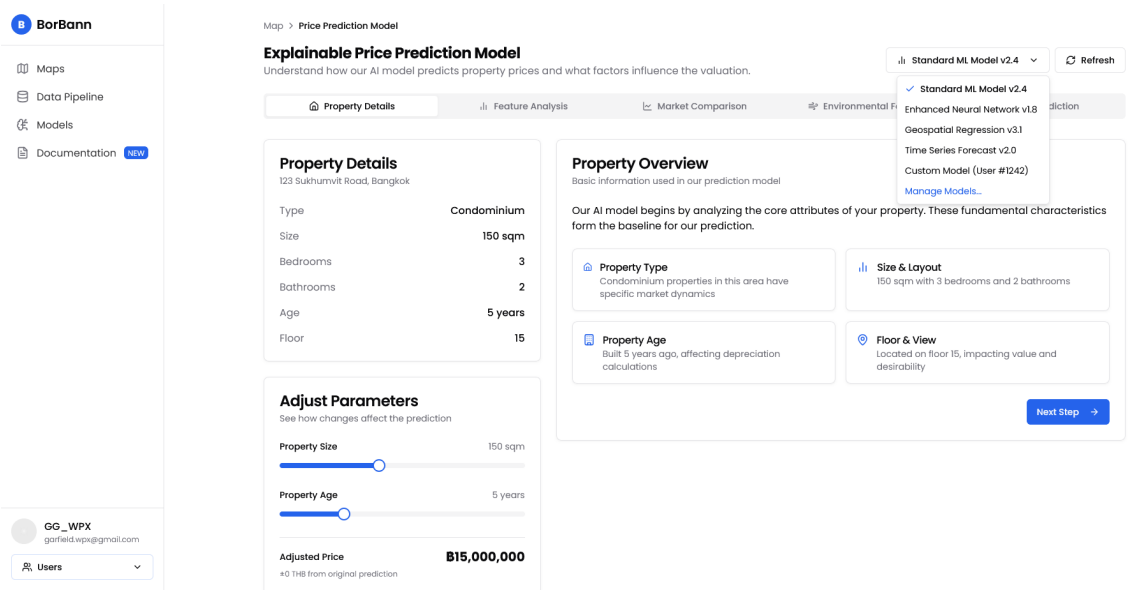

Figure 3.21: Explainable Price Prediction - Property Overview

Figure 3.21 displays the initial analysis of property attributes in the prediction model. The interface includes interactive parameter sliders that show how changes to property characteristics affect the predicted price.

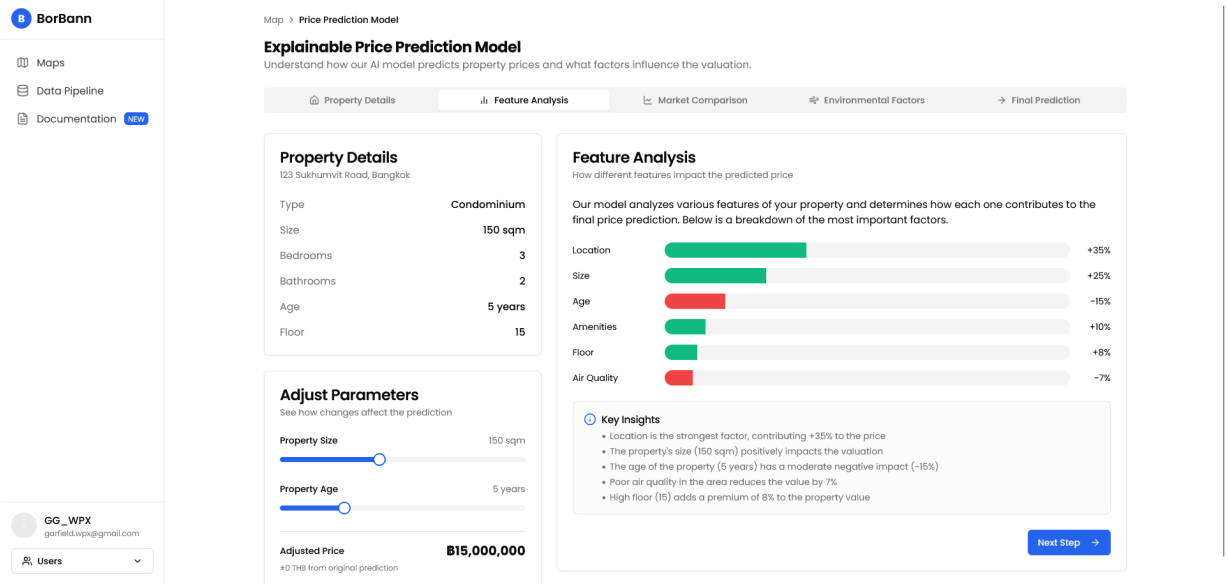

Figure 3.22: Explainable Price Prediction - Feature Importance Visualization

Figure 3.22 shows the contribution of different property features to the predicted price. The visualization uses color-coded bars to differentiate positive and negative factors with percentage impact values.

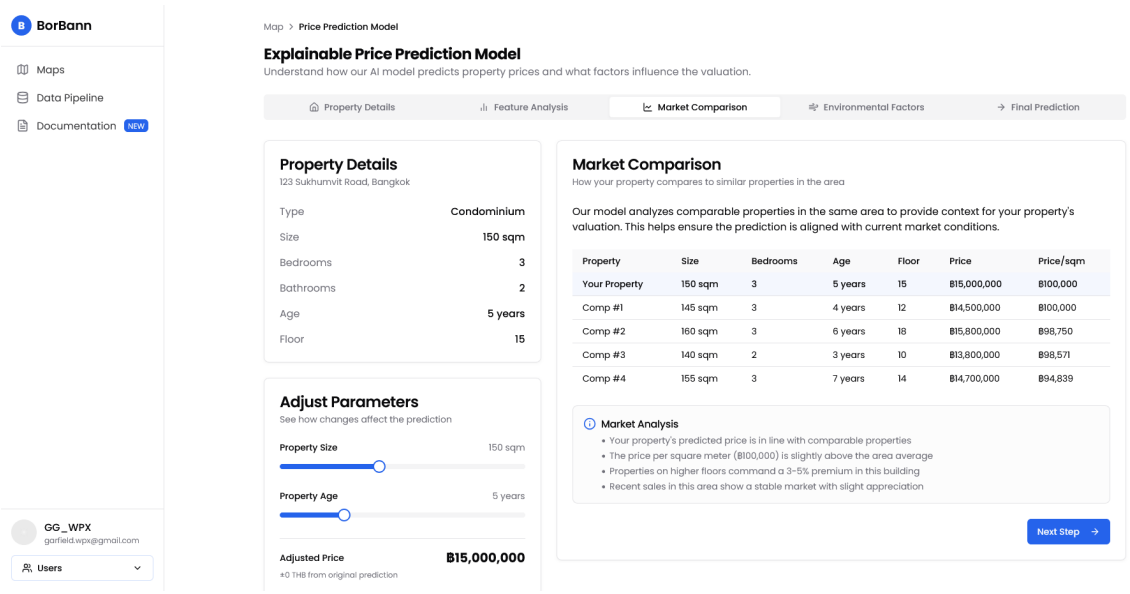

Figure 3.23: Explainable Price Prediction - Comparative Market Analysis

Figure 3.23 presents a comparison of the subject property against similar properties in the area. The table highlights key attributes and pricing metrics with supporting analysis of market positioning.

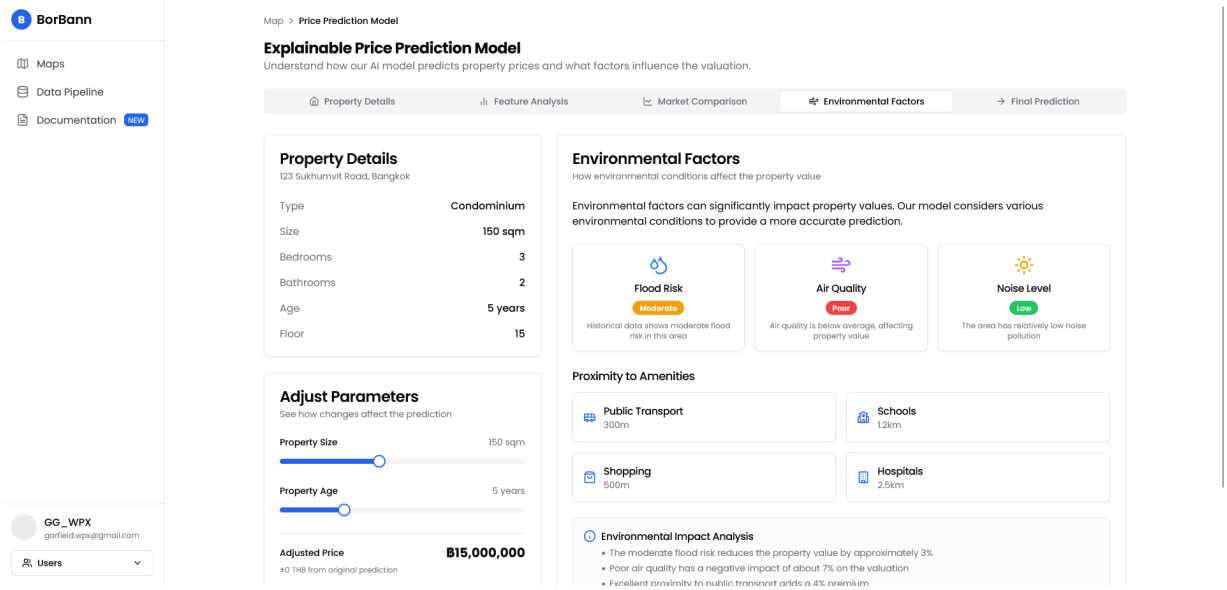

Figure 3.24: Explainable Price Prediction - Environmental Impact Analysis

Figure 3.24 analyzes environmental conditions and nearby amenities affecting property value. The interface displays flood risk, air quality, and proximity to key facilities with quantified impact on valuation.

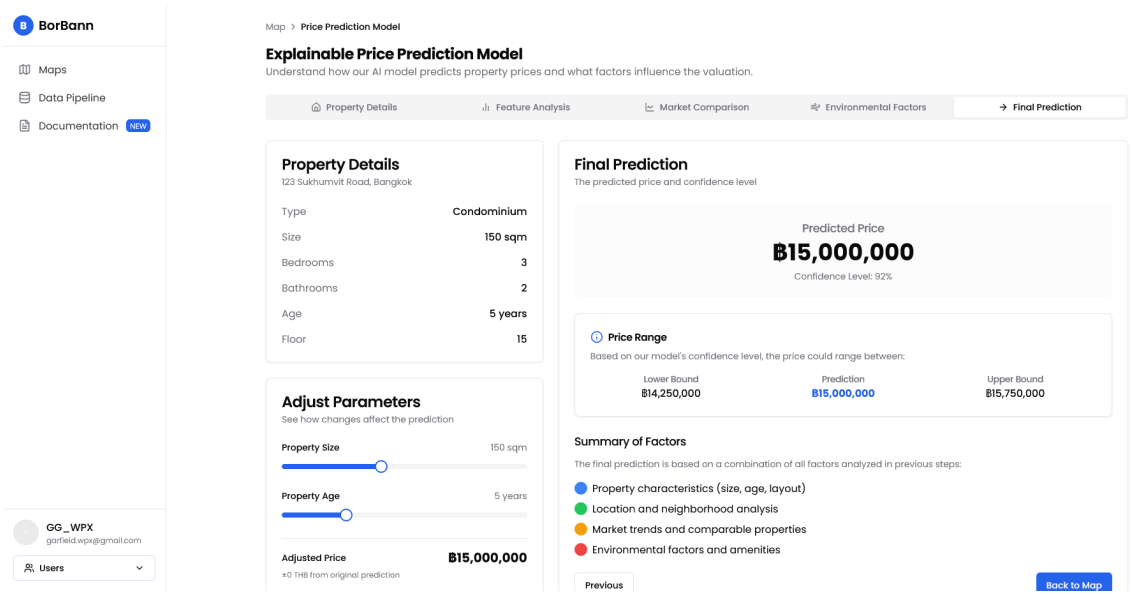

Figure 3.25: Explainable Price Prediction - Final Valuation with Confidence Range

Figure 3.25 shows the final price prediction with confidence metrics and range indicators. The summary section explains how the valuation synthesizes insights from all analytical components.

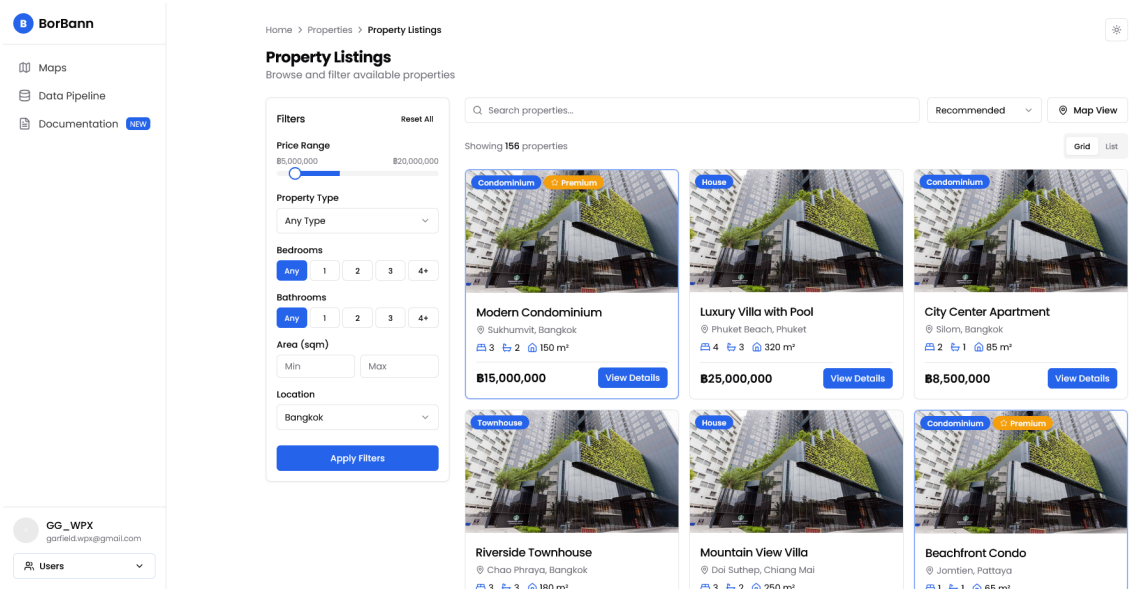

Figure 3.26: Property Listings Page

Figure 3.26 display property listings page with all listing that follow the filter tab the right.

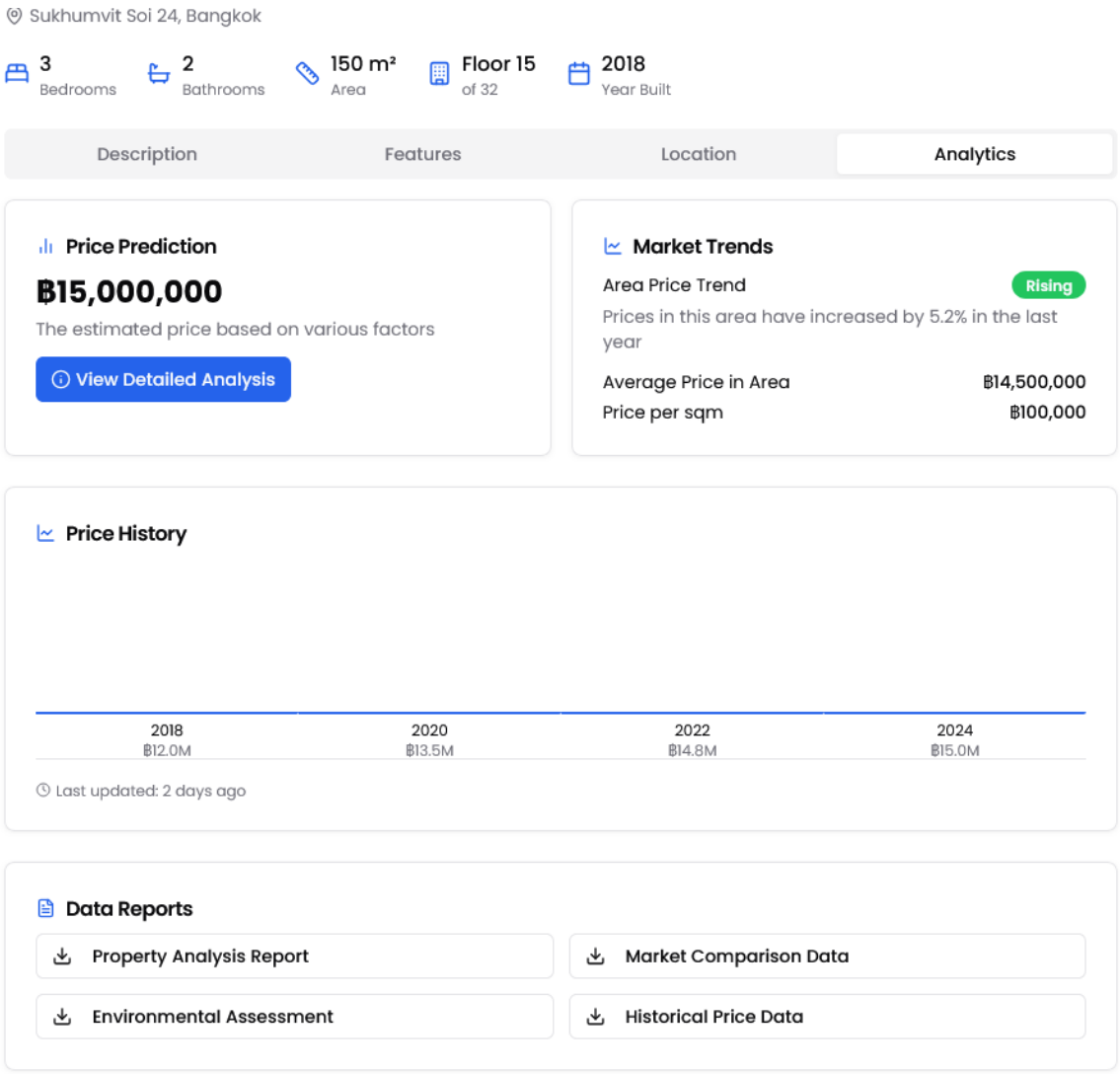

Figure 3.27: Property Listings Page - Analytic Tab

Figure 3.27 shows the analytic tab that show local context analytic of that specific listing.

Chapter 4

Software Architecture Design

4.1 Sequence Diagram

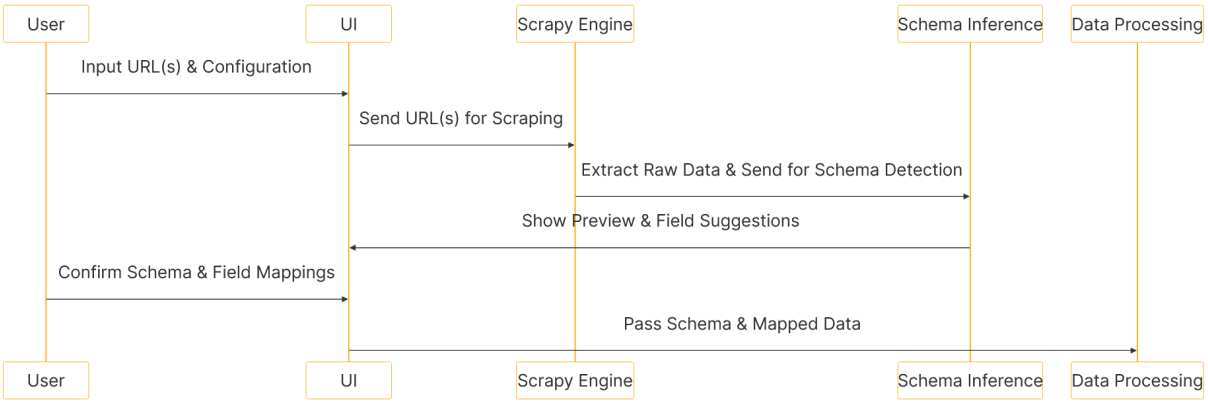

Figure 4.1: Sequence Diagram of Customizable Automated Data Integration Pipeline

The sequence diagram show the flow of interactions between key components of the customizable automated data integration pipeline. The process begins with the **User**, who inputs URLs and configuration details via the **UI**. The **UI** sends the provided URLs to the **Scrapy Engine** for scraping.

Once the data is extracted, the **Scrapy Engine** forwards the raw data to the **Schema Inference** module for automatic schema detection. The **Schema Inference** module analyzes the data and returns a preview, along with field mapping suggestions, to the **UI**. The **User** then reviews the suggested schema and field mappings in the **UI**.

Upon confirmation by the **User**, the **UI** passes the finalized schema and mapped data to the **Data Processing** pipeline, where further processing and integration of the collected data occur.

4.2 AI Component

The BorBann platform has AI components that enhance its analytical capabilities and user experience. These components work together to provide build a real estate data platform adapt to the Thai market context. Each component integrate with each other to form a comprehensive analytics system:

- The Data Integration Pipeline feeds data to the Local Contextual Analytics system
- Local Contextual Analytics provides features to the Explainable Price Prediction Model
- The Retraining Model uses data from the pipeline to create customized predictions
- All components share a common data model that enables smooth information flow

Explainable Price Prediction Model

The price prediction model delivers property price predictions with clear explanations of contributing factors.

Model Architecture

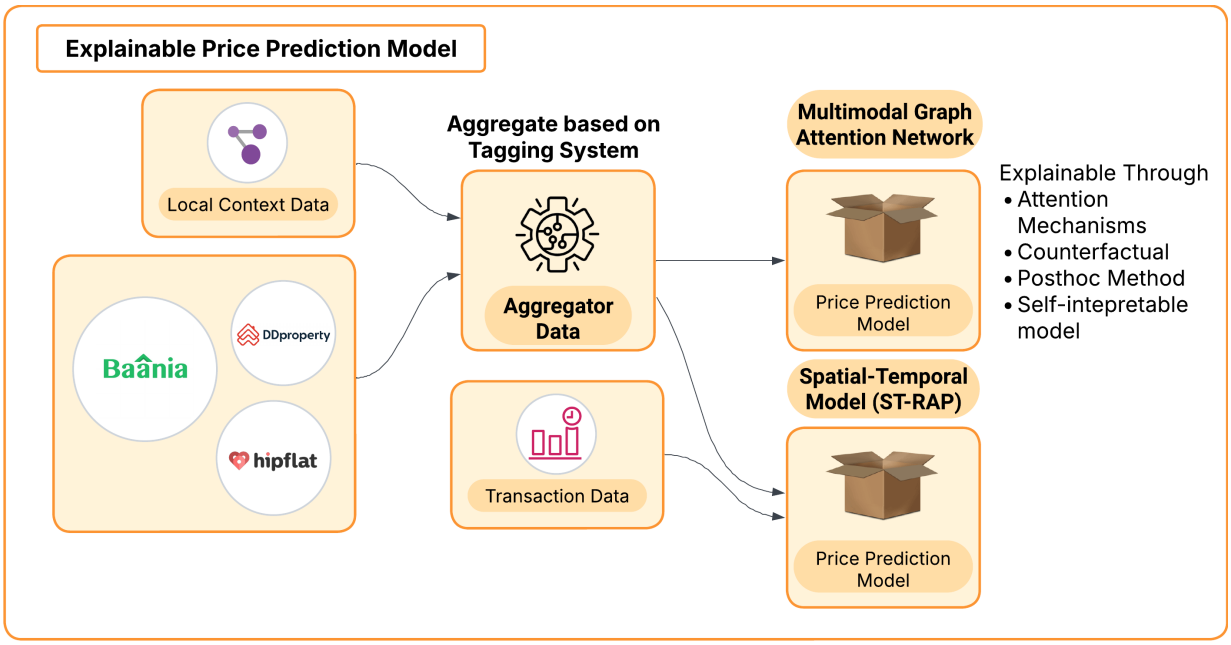

Figure 4.2: Explainable Price Prediction Data Flow

Figure 4.2 illustrates the architecture of the Explainable Price Prediction Model. The diagram shows how data flows from multiple sources (Baania, DDproperty, hipflat) and local context data into an Aggregator component that organizes information based on a tagging system. The aggregated data then feeds into two distinct prediction models:

a Multimodal Graph Attention Network and a Spatial-Temporal Model (ST-RAP). The explainability features are highlighted on the right, showing how the model provides transparency through attention mechanisms, counterfactual analysis, post-hoc methods, and self-interpretable modeling approaches.

- **Dual Modeling Approach:**
  - *Tree-based Model:* With tree-based gradient boosting models (XGBoost[4]).
  - *Multimodal Graph Attention Network:* With Multimodal Graph Attention Network[5] that is multi-modal model which can handle large graph.
  - *Spatial-Temporal Model:* Using ST-RAP[6] for facilities-property graph input and time series of real estate pricing data

Explainability Framework

- **In-model Attention:** With attention mechanism, we can look into weighted contributions of nodes, edges, or modalities
- **Post-hoc Explanation Module:** Uses KernelSHAP[7] (Algorithm to approximate SHapley Additive exPlanations) value calculation for feature attribution.
- **Self-Interpretable Components:** Extracts rules from complex models and implements decision tree surrogate models that approximate complex model behavior

User Interface Components

- **Interactive Visualizations:** Feature importance charts, and price trend projections with confidence intervals
- **Explanation Generation:** Natural language generation of price explanations and highlighting of key value drivers

Local Contextual Analytics

This component analyzes environmental conditions and proximity factors to evaluate property context and risk:

Model Architecture

Figure 4.3 depicts the Local Contextual Analytics system. The diagram shows various data sources including GISTDA, Bangkok Metropolitan Administration, news outlets, and property listing platforms (Baania, hipflat) feeding into an Aggregator Service. This service collects different types of data through web scraping, API fetching, and dataset downloads. The collected data undergoes preprocessing with outlier detection, value imputation, and feature engineering before being stored in specialized databases (Vector, Time-Series,

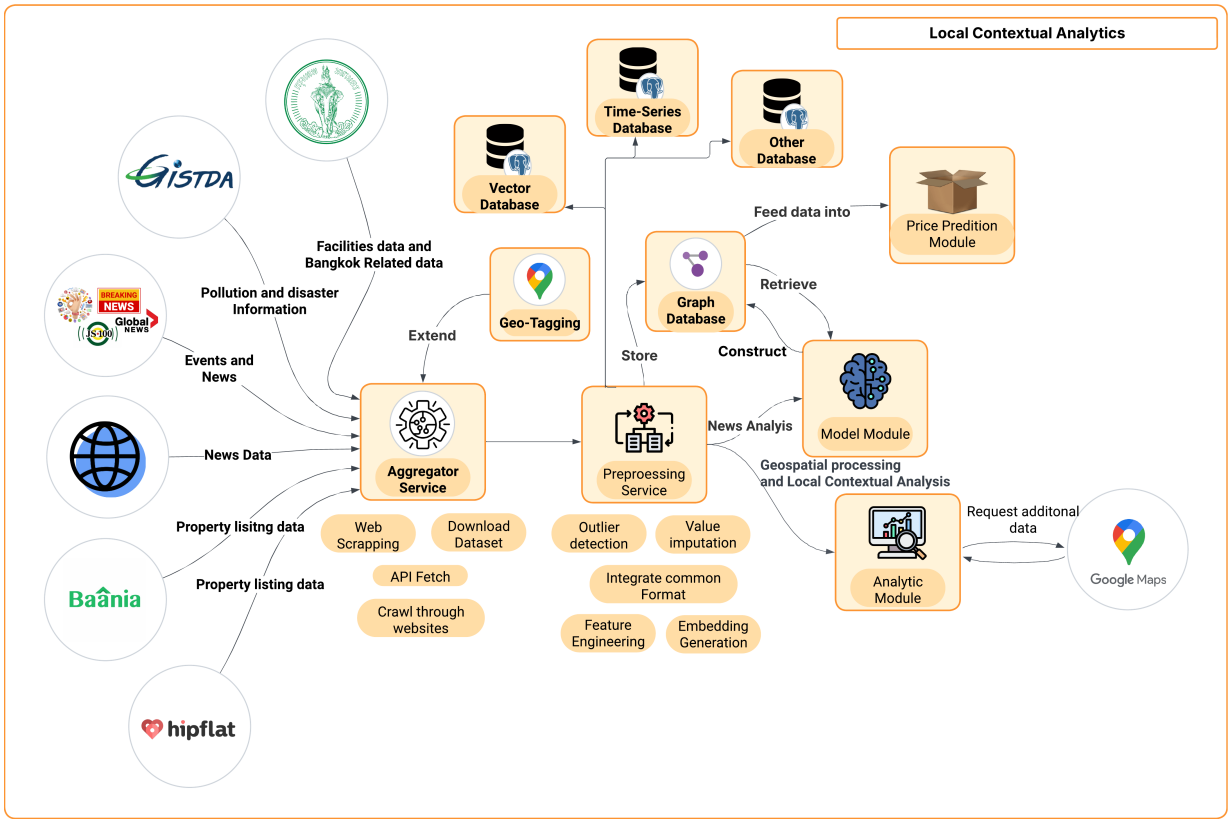

Figure 4.3: Local Contextual Analytic Data Flow

Graph). The Model Module and Analytic Module process this information to provide geospatial analysis and local contextual insights that ultimately feed into the Price Prediction Module.

- **Graph Database:** Stores heterogeneous data in Neo4J or other knowledge graph database
- **Vector Database:** Stores embedding from preprocessing unit within PostGreSQL with pgvector that will provide vector database capabilities
- **Timeseries Database:** Stores time-series data withtin Timescale
- **Relational Database:** Stores relational data in PostgreSQL
- **NoSQL Database:** Stores unstructured data within MongoDB
- **Model Module:**
  - Heterogeneous Graph Construction using k-NN or GCN
  - GCN models for spatial relationships
  - NLP models for news analysis
- **Price Prediction Module:** Integrated model consuming all processed features

Analytics Capabilities

- **Climate and Environmental Analysis:** Evaluates climate risk, pollution levels, and disaster assessment
- **Proximity Analysis:** Evaluates nearby locations and facilities, analyzing relationships between facilities
- **News Integration:** Incorporates local news into property assessment

Customizable Automated Data Integration Pipeline

This pipeline enables non-technical users to connect any data source into a unified system:

LLM Module

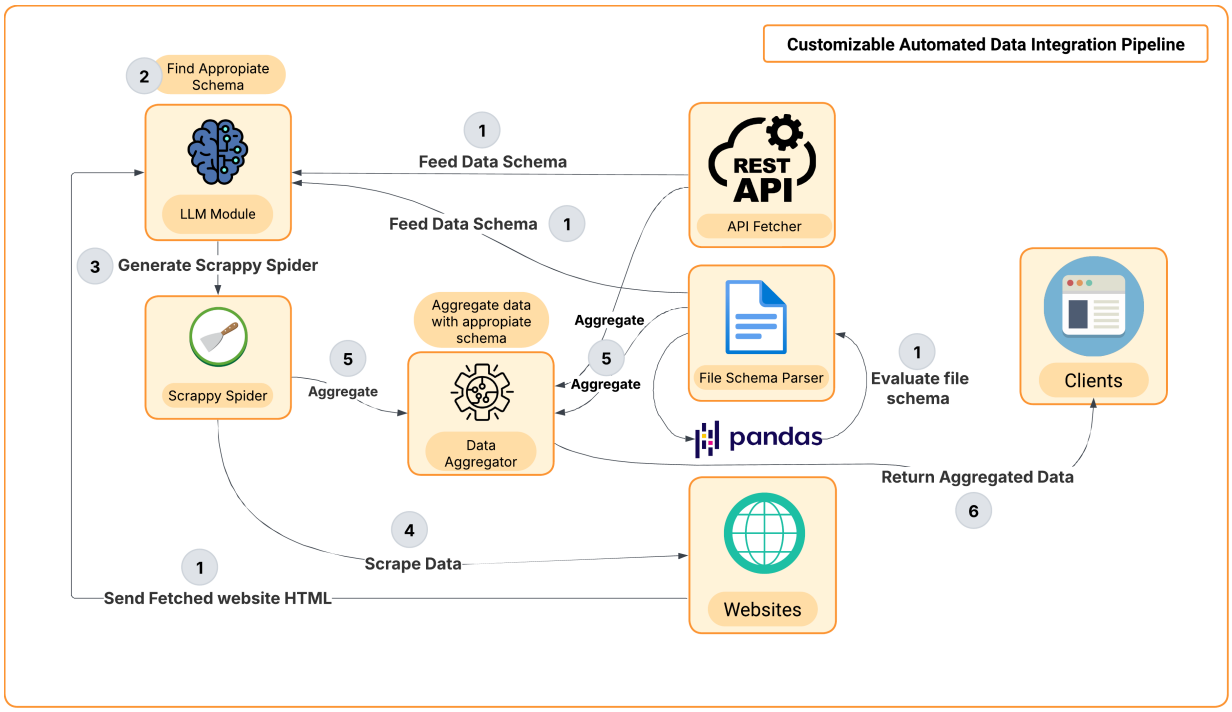

Figure 4.4: Customizable Automated Data Pipeline Data Flow

Figure 4.4 presents the Customizable Automated Data Integration Pipeline. The workflow begins with data sources (API Fetcher, File Schema Parser, Websites) being evaluated. An LLM Module analyzes these sources to find appropriate schemas (step 2), then generates Scrappy spiders for web scraping (step 3). The spiders extract data from websites (step 4), while the Data Aggregator combines information from all sources (step 5). Finally, the aggregated data is returned to clients (step 6). This pipeline enables non-technical users to integrate diverse data sources through an automated, LLM process.

- Analyzes data sources to find appropriate schemas

- For websites, generates Scrappy spider configurations automatically
- Bridges the gap between unstructured and structured data

Data Processing

- **Scrappy Spider:** Dynamically generated web scrapers based on LLM analysis that extract targeted data from websites according to the determined schema
- **Data Aggregator:** Central component that combines data from all sources, harmonizes different schemas into a unified structure, and uses appropriate schema mapping determined by the LLM

Retraining Model with Pipeline Data

This component allows users to create custom prediction models by combining their pipeline data with platform data:

Technical Implementation

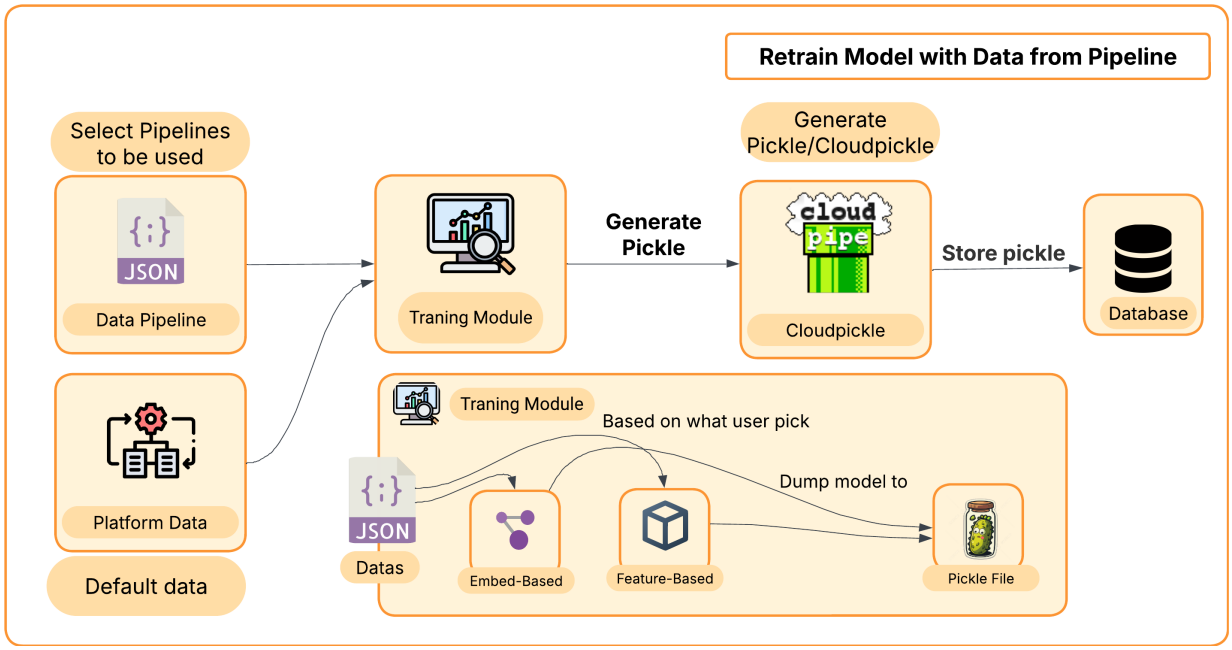

Figure 4.5: Retrain Model with Data from Pipeline Data Flow

Figure 4.5 shows the Model Retraining Pipeline. The diagram illustrates how users can select pipelines and combine them with platform data to train custom models. The Training Module processes this data and allows users to choose between embed-based or feature-based modeling approaches. The system generates pickle/cloudpickle files of the trained models, which are then stored in a database for future use. This component empowers users to create specialized prediction models tailored to their specific data and requirements.

- **Training Engine:** Implements automated training engine that rely on embedding-based models to avoid problem with unmatch input feature and tree-based models (for structured data), and builds parameter tuning system using random search if user doesn't specify ones.
- **Model Serialization:** Creates versioned model files for persistence and stores serialized models in pickle format in database with metadata
- **Deployment Framework:** Provides APIs for model inference, enables batch prediction, and implements monitoring for model drift detection

Chapter 5

AI Component Design

This chapter describes how Artificial Intelligence (AI) components are designed and integrated into the overall system. Each section starts with a clear objective and provides guidance on analysis, design, implementation, and evaluation of AI modules.

5.1 Business Context and AI Integration

Figure 5.1 shows the overview of the system workflow and how each component works together. It illustrates that AI will be used in four components: Data Integration Pipeline, Explainable Price Prediction, Property and Neighborhood Insight, and Pricing Explanation.

For each component, I will explain why using AI is both feasible and necessary.

1. Data Integration Pipeline

Using AI is suitable in this case because content on websites varies by type of information and other elements, so AI needs to be implemented to help parse and understand the context and extract information.

When aggregating multiple data sources, differences in data schemas present a significant challenge. We need to establish a centralized data schema, and AI can help users accomplish this more easily by suggesting related fields.

For web scraping, the problem is complex and constantly changing because website structures vary across multiple sites and are difficult to parse with fixed rules. For file and API sources, the challenge is less complex since we can extract data from files and fetch data from API endpoints directly. However, complexity increases when combining these sources into one unified pipeline.

We can accept some incompleteness in this module. Missing fields in the aggregated data will not significantly affect the analysis when working with large volumes of data.

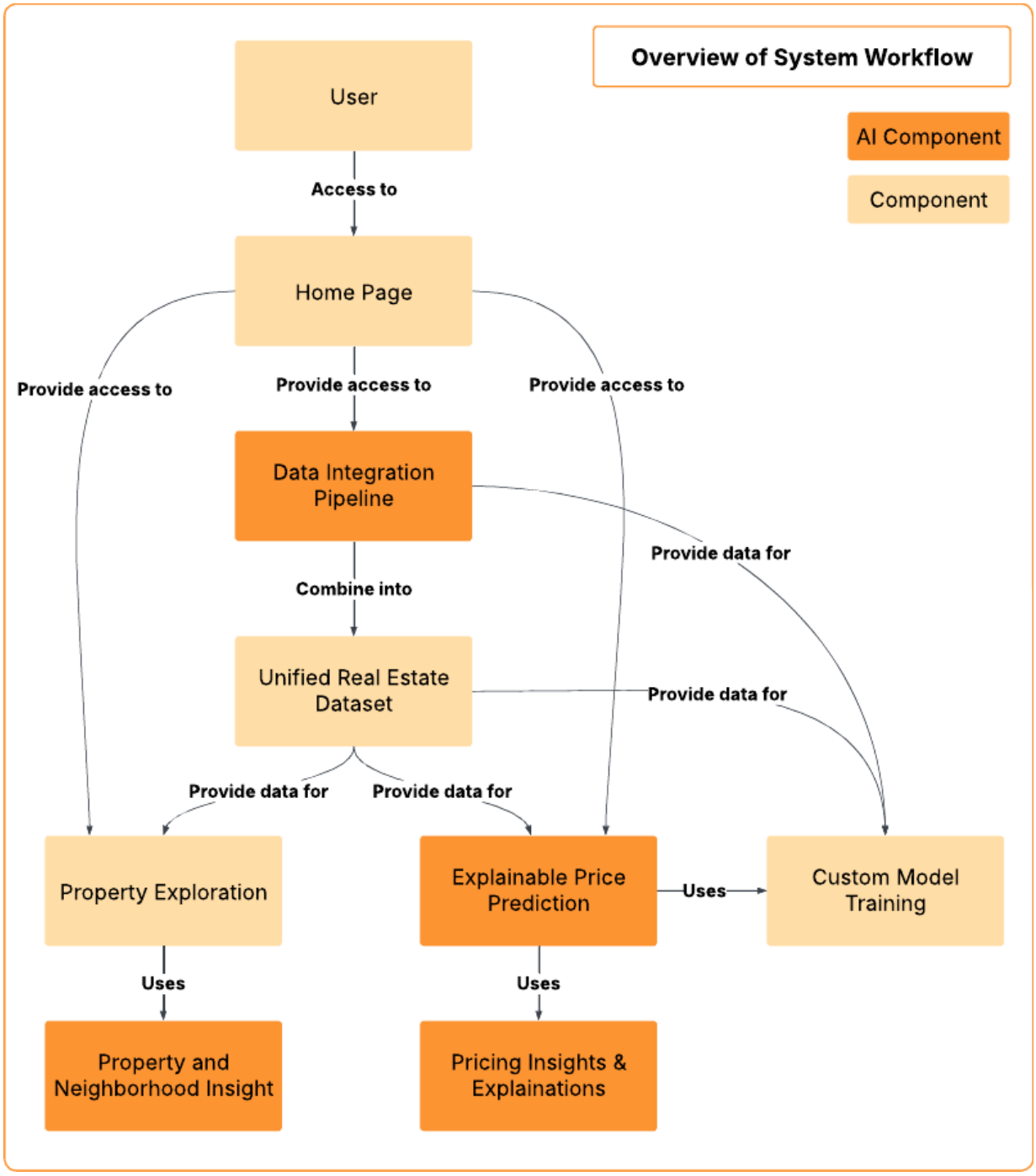

Figure 5.1: Overview system workflow

2. Explainable Price Prediction

AI is appropriate for price prediction because housing prices are influenced by numerous factors that interact in complex, non-linear ways. Traditional rule-based systems cannot effectively capture these intricate relationships.

The housing market constantly evolves with changing economic conditions, buyer preferences, and seasonal variations. AI models can adapt to these shifts by identifying new patterns in the data that might not be obvious to human analysts.

We can accept predictions that are not 100% accurate since real estate valuation inherently involves some uncertainty. However, by using explainable AI approaches, we can provide confidence intervals and explain the key factors influencing each prediction, making the system valuable even with some margin of error.

3. Property and Neighborhood Insight

AI is well-suited for generating property and neighborhood insights because this requires analyzing diverse data types including text descriptions, images, geographic information, and numerical data. The relationships between these varied data sources are complex and difficult to codify with traditional rules.

The characteristics that make a neighborhood desirable change over time and vary across different buyer segments. AI can identify emerging trends and personalized insights that static analysis would miss.

Perfect completeness is not essential, as providing valuable insights on the most significant factors affecting property desirability is more important than capturing every minor detail. Users benefit from focused, relevant information rather than exhaustive analysis.

4. Pricing Explanation

Using AI for pricing explanation is appropriate because explaining property valuations involves communicating complex relationships between numerous factors in an accessible way. These explanations need to adapt to each property's unique characteristics and the specific market context.

The relative importance of different pricing factors changes across markets and over time. AI can generate contextual explanations that reflect these dynamics rather than relying on fixed templates.

While we can accept explanations that might not cover every possible factor influencing price, the system provides significant value by highlighting the most important considerations and presenting them in a clear, understandable format for users.

5.2 Goal Hierarchy

This section outlines the hierarchical structure of goals for our real estate valuation system, beginning with organizational objectives and flowing down to specific AI model goals. For each level, We provide clear metrics to measure success.

Organizational Goals

The primary organizational goals for this AI system are:

- To establish a trusted platform for accurate and transparent real estate valuations
- To build a reputation for innovation in applying AI to real estate challenges

Success at the organizational level will be measured through:

- Recognition through awards

System Goals

At the system level, our goals are:

- To integrate diverse real estate data sources into a unified, high-quality dataset
- To generate accurate property valuations with clear explanations for price factors
- To provide insightful property and neighborhood analysis to inform decision-making
- To deliver a reliable, responsive user experience across different devices and user types

System success metrics include:

- System uptime and response time statistics
- Data completeness and quality indicators across different property types and regions
- System error rates and exception handling effectiveness
- System scalability under increasing data volumes and user numbers

User Goals

For our users, the goals are:

- To receive insightful property valuations they can trust for decision-making
- To understand the key factors influencing property prices in specific markets
- To gain insights about properties and neighborhoods beyond basic statistics
- To save time in researching and analyzing real estate opportunities

User success metrics will be assessed through:

- User satisfaction surveys and net promoter scores

AI Model Goals

At the most specific level, our AI model goals are:

- To accurately predict property prices with clearly quantified confidence levels
- To identify and weigh the most influential factors affecting property values
- To provide explanations for predictions that are both technically sound and accessible to non-technical users

AI model performance will be measured through:

- Statistical accuracy metrics including mean absolute error, root mean squared error, and R-squared value compared to actual transaction prices
- Explanation quality assessed through user comprehension testing
- Comparative performance against benchmark models and traditional valuation methods
- Drift detection metrics to identify when model retraining is necessary
- Computational efficiency metrics including inference time and resource utilization

5.3 Task Requirements Analysis Using AI Canvas

This section presents a detailed analysis of our AI system's task requirements using the AI Canvas methodology. This approach helps us clearly articulate what each AI component should accomplish, how it will operate technically, and the conditions under which it will function.

AI Task Requirements

For each AI component in our system, we analyze three key dimensions:

- **Requirements (REQ):** The specific objectives and functions each AI component must fulfill
- **Specifications (SPEC):** The technical approach and methods the AI will employ
- **Environment (ENV):** The operational conditions, constraints, and context in which the AI will function

1. Data Integration Pipeline

Requirements (REQ): The AI component must extract structured data from diverse sources including property listing websites, government databases, and unstructured documents. It needs to identify property characteristics, pricing information, location data, and temporal attributes across varying formats. The system must normalize extracted data into a unified schema.

Specifications (SPEC): This component will implement a hybrid approach combining rule-based extraction for standardized sources and deep learning models for unstructured content. Named entity recognition models will identify property attributes within text, while transformer-based architectures will handle context-dependent extraction tasks.

Environment (ENV): The component will operate in a data environment characterized by high heterogeneity across sources, frequent changes in website structures, and varying data quality. It must process large volumes of daily updates while maintaining extraction accuracy. The system requires periodic retraining as new data sources are integrated and reasonable human oversight for exception handling of complex cases that fall outside standard patterns.

2. Explainable Price Prediction

Requirements (REQ): This AI component must generate accurate property valuation estimates with quantified confidence intervals. It needs to identify key value drivers specific to each property, adapt to regional market dynamics, and provide predictions that remain reliable even with incomplete input data. The system must detect when it lacks sufficient information for reliable prediction and communicate these limitations transparently.

Specifications (SPEC): The prediction engine will utilize an ensemble approach combining gradient-boosted decision trees for tabular data with graph neural networks to capture neighborhood relationships and spatial dependencies. SHAP values and counterfactual explanations will provide feature importance analysis.

Environment (ENV): This component operates within a complex market environment.

3. Property and Neighborhood Insight

Requirements (REQ): The AI must analyze multilayered neighborhood data to identify significant patterns and trends relevant to property valuation. It needs to generate insights about local amenities, infrastructure developments, community characteristics, and future growth potential.

Specifications (SPEC): This component will utilize multimodal analysis techniques combining geospatial machine learning for location-based features with natural language processing for textual descriptions and sentiment analysis. The system will implement clustering algorithms to identify comparable neighborhoods and temporal modeling to detect emerging trends.

Environment (ENV): The insight generation occurs in an information-rich but fragmented environment with data spanning multiple domains including transportation, education, commerce, and safety. The component must operate with varying data availability across neighborhoods and maintain cultural sensitivity when analyzing community characteristics. It needs to distinguish between transient and persistent neighborhood attributes while balancing detail with relevance in its output.

4. Pricing Explanation

Requirements (REQ): This component must translate complex valuation models into clear, intuitive explanations. It needs to communicate uncertainty appropriately, highlight the most influential factors affecting a specific property's value, and provide comparative analysis with similar properties. The system should generate explanations that help users understand market dynamics without overwhelming them with technical details.

Specifications (SPEC): The explanation system will implement a natural language generation pipeline built on a large language model fine-tuned for real estate domain knowledge. It will incorporate techniques from the field of explainable AI including feature attribution visualization, importance ranking. The system will use templates for consistent structure while leveraging dynamic content generation for property-specific insights.

Environment (ENV): This component operates at the interface between technical prediction systems and non-technical users.

AI Canvas Development

1. Data schema mapping

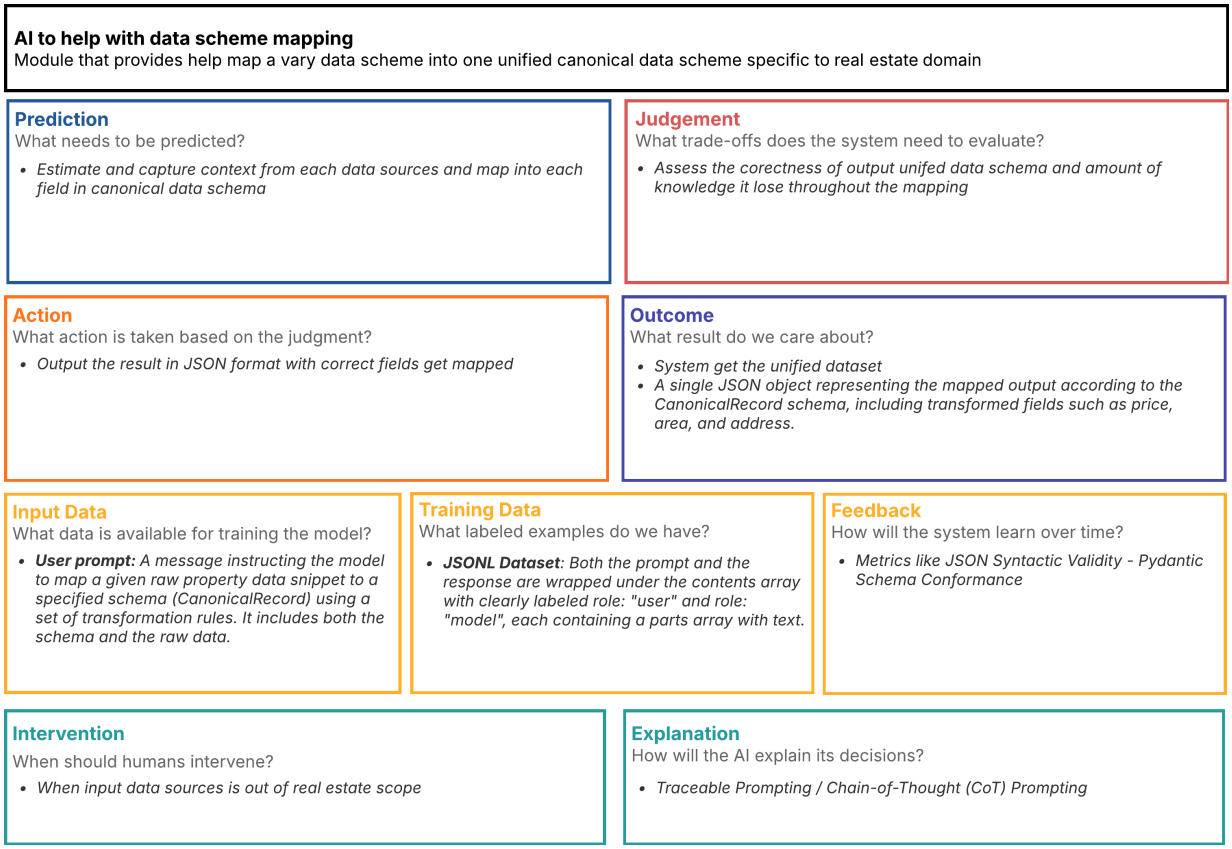

Figure 5.2: Data schema mapping AI Canvas

Figure 5.2 presents an AI Canvas diagram for a Real Estate Data Schema Mapping system. This strategic planning tool articulates how artificial intelligence creates value through structured components that guide the implementation of a unified canonical data schema for real estate information.

The AI Canvas comprises eight interconnected sections that collectively define the system's purpose and operation. The Prediction section establishes the core functionality: estimating and capturing context from each data source and mapping it into each field in the canonical data schema. This works in concert with the Judgment section, which articulates the critical trade-offs the system must evaluate, focusing on assessing the correctness of the output unified data schema and measuring the amount of knowledge potentially lost throughout the mapping process.

The Action section defines how the system's outputs are translated into tangible steps, outputting the results in JSON format with correctly mapped fields. These actions lead to the Outcome section, which clarifies the ultimate value proposition: generating a unified dataset represented as a single JSON object conforming to the CanonicalRecord schema, including transformed fields such as price, area, and address.

The Input Data section catalogues the available information sources: user prompts containing instructions for mapping raw property data snippets to a specified schema (CanonicalRecord) using transformation rules, including both the schema specifications and the raw data itself. Complementing this, the Training Data section defines the labeled examples powering the model: JSONL datasets where both prompts and responses are wrapped in contents arrays with clearly labeled roles ("user" and "model"), each containing parts arrays with text.

The Feedback section outlines how the model will learn over time by tracking metrics like JSON Syntactic Validity and Pydantic Schema Conformance. The Intervention section establishes boundaries for human oversight, calling for expert involvement when input data sources fall outside the real estate scope. The Explanation section details the technical approaches for transparency: Traceable Prompting and Chain-of-Thought (CoT) Prompting methodologies to provide insight into the system's decision-making processes.

2. Explainable Price Predition

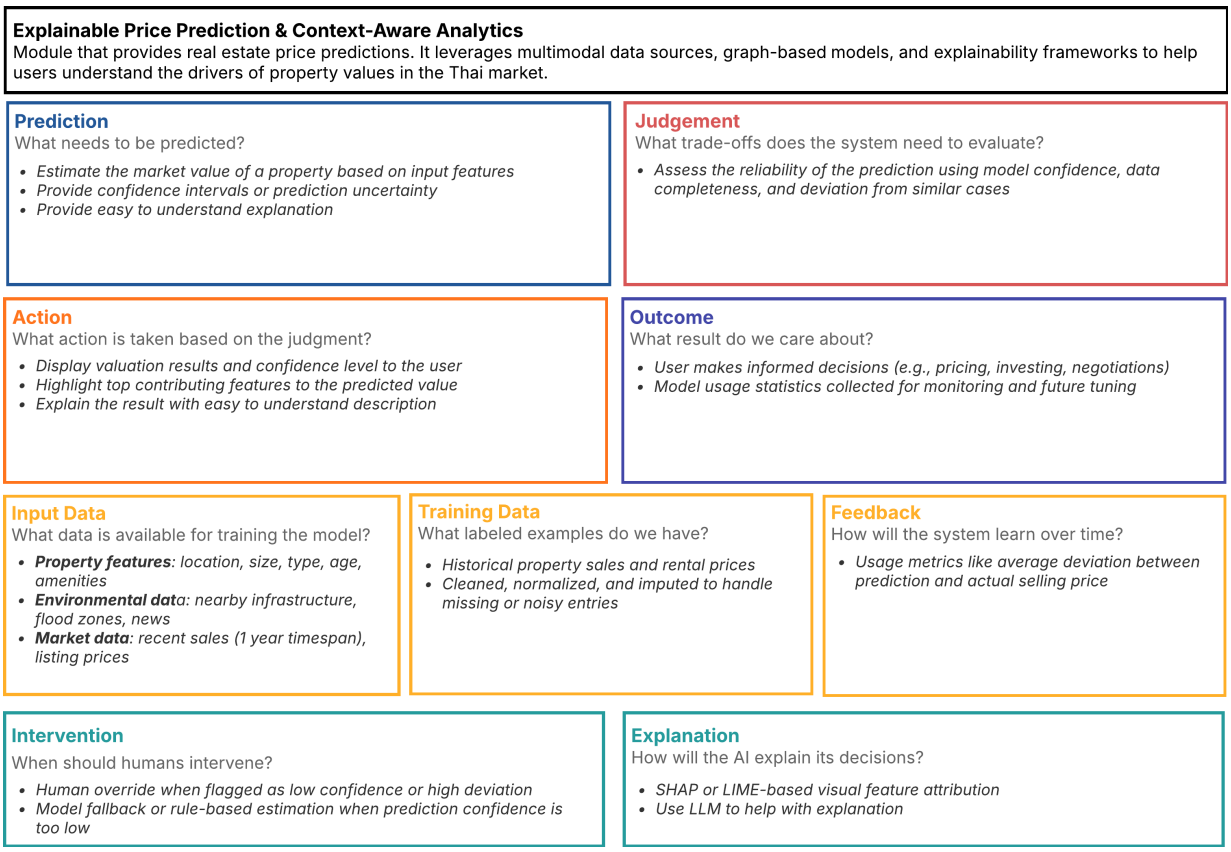

Figure 5.3: Explainable Price Predition AI Canvas

Figure 5.3 presents an AI Canvas diagram for an Explainable Price Prediction & Context-Aware Analytics system tailored to the Thai real estate market. This strategic planning tool articulates how artificial intelligence creates value through structured components that guide implementation.

The AI Canvas comprises eight interconnected sections that collectively define the system's purpose and operation. The Prediction section establishes the core functionality: estimating property market values based on input features, providing confidence intervals to quantify uncertainty, and delivering accessible explanations to users. This works in concert with the Judgment section, which articulates the critical trade-offs the system must evaluate, focusing on assessing prediction reliability through model confidence metrics, data completeness evaluation, and deviation analysis from comparable properties.

The Action section defines how the system's outputs are translated into tangible steps, displaying valuation results with confidence levels, highlighting the top contributing features to predicted values, and explaining results through user-friendly descriptions. These actions lead to the Outcome section, which clarifies the ultimate value proposition: enabling users to make informed real estate decisions across pricing, investing, and negotiations, while simultaneously collecting usage statistics for ongoing system optimization.

The Input Data section catalogues the available information sources: property features including location, size, type, age, and amenities; environmental data encompassing infrastructure, flood zones, and news; and market data covering recent sales and current listings. Complementing this, the Training Data section defines the labeled examples powering the model: historical sales and rental prices that have undergone cleaning, normalization, and imputation processes to handle data quality issues.

The Feedback section outlines how the model will learn over time by tracking metrics like average deviation between predictions and actual selling prices. The Intervention section establishes boundaries for human oversight, calling for expert involvement when predictions show low confidence or high deviation, while implementing fallback mechanisms when prediction certainty falls below acceptable thresholds. The Explanation section details the technical approaches for transparency: SHAP or LIME-based visual feature attribution combined with Large Language Models to generate intuitive explanations.

5.4 User Experience Design with AI

This section explains how AI is used to improve the user experience in our real estate system. AI is not just a background tool—it helps users by doing tasks, adding helpful info, giving smart suggestions, and adjusting to user needs. These features are built into key parts of the system, like data handling, neighborhood analysis, and price prediction. The design focuses on being clear, easy to use, and useful for different types of users.

Automated Pipelines

The pipeline system in BorBann is not just a backend tool. It's a dynamic, AI interface that enables even non-technical users to build, manage, and leverage data workflows for training personalized real estate models.

The system is designed using an **Automate** style that automatically handles repetitive technical tasks—like schema inference and spider generation—but still gives users the power to manually adjust the process.

1. Pipeline Creation and Management

Users start by selecting data sources—like websites, files, or APIs—and use a no-code interface to configure their pipeline. The system generates scraping rules using an LLM and recommends schema alignments automatically.

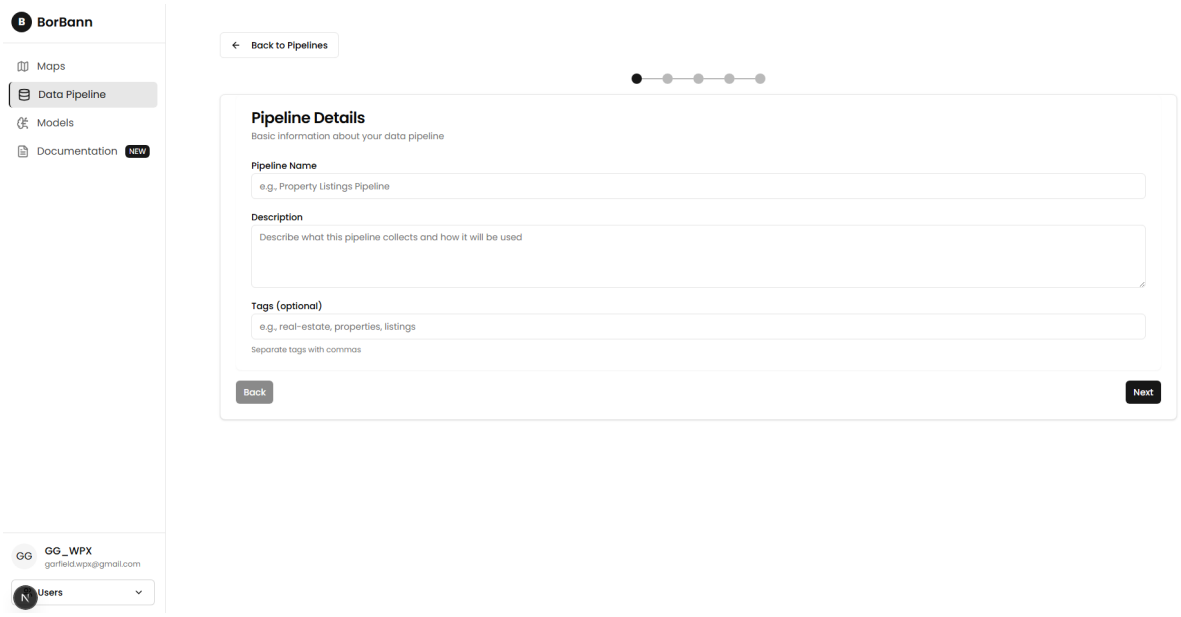

Figure 5.4: Pipeline Creation Interface

Figures 5.4 and 5.5 show how users input data source URLs and set extraction options with help from the AI.

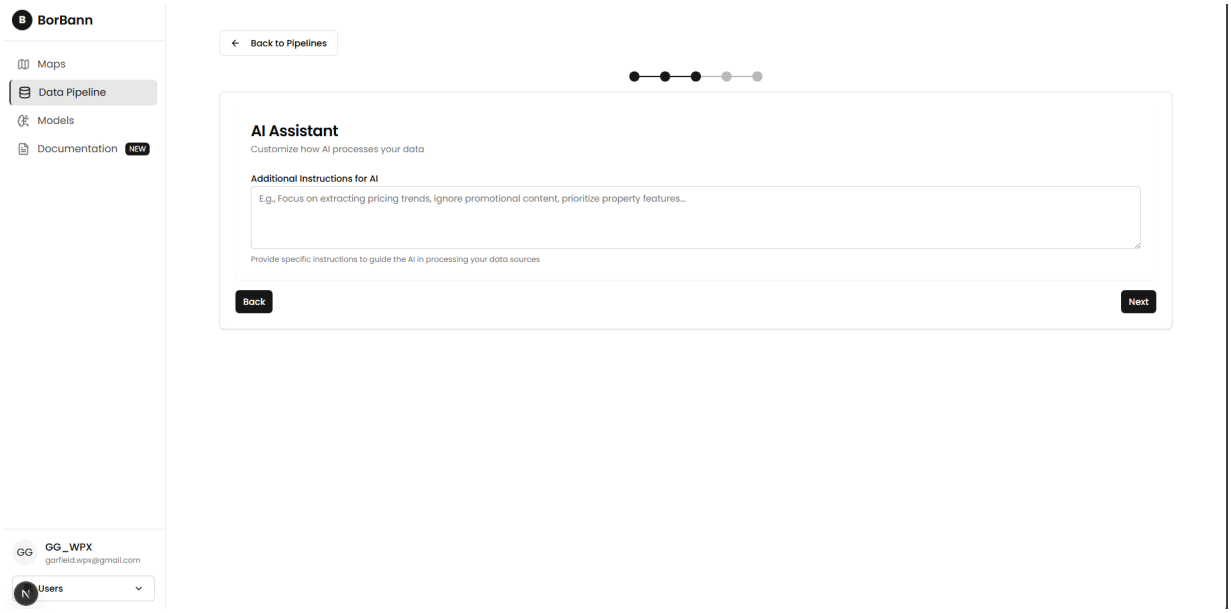

Figure 5.5: Pipeline Creation Interface – Additional Prompt

2. Field Customization and Schema Annotation

After setting up the source, users can view, modify, or remove fields detected by the system. The interface visually maps fields across sources, includes data type validation, and supports custom field creation via formulas.

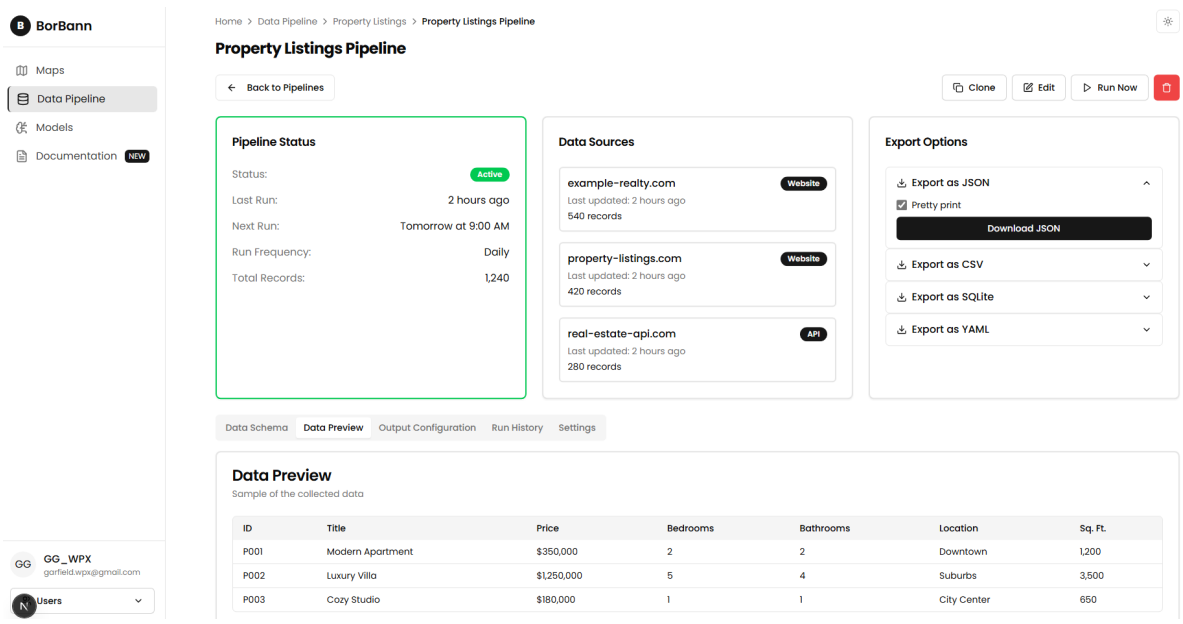

Figure 5.6: Field Management Interface

Figure 5.6 shows the functionalities to customize the schema and data source integration.

3. Pipeline Monitoring and Status Overview

Users can view a dashboard that summarizes the status of each pipeline—categorized as Active, Paused, Failed, or Completed. Each pipeline card is annotated with insights.

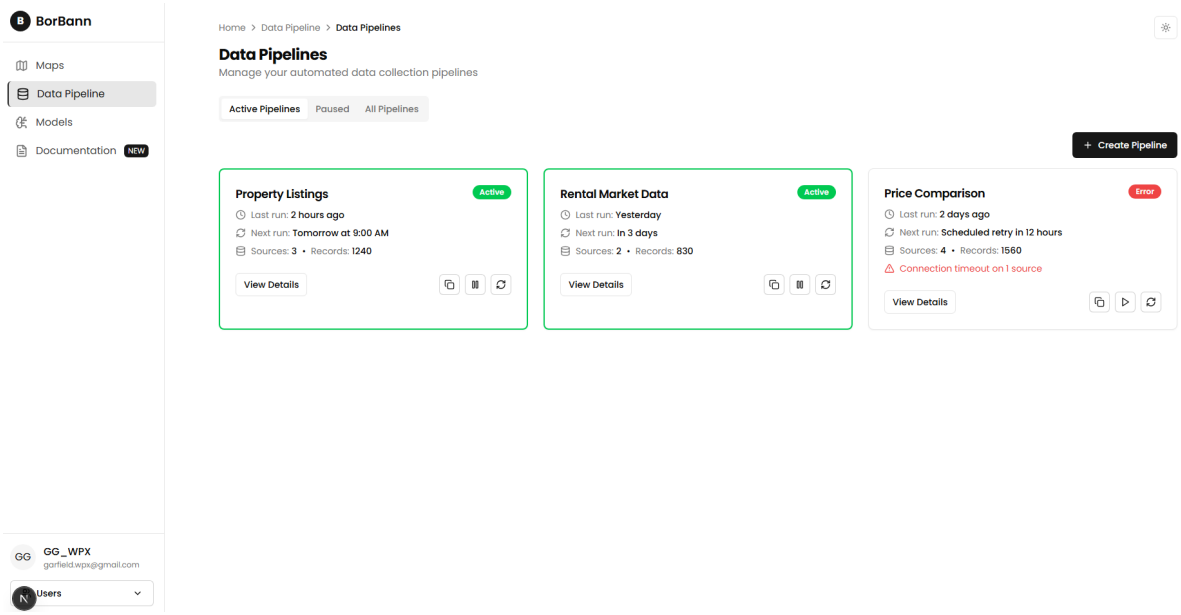

Figure 5.7: Pipeline Dashboard Interface

Figure 5.7 shows how the interface represents pipeline statuses; successful pipelines are highlighted with green-bordered cards.

Neighborhood Insight

The Neighborhood Insight feature uses various APIs and tools to analyze environmental conditions, nearby facilities, local amenities, and even news sentiment. The goal is to support smarter decisions by making the neighborhood story as transparent and rich as the property details.

This feature uses **Annotate**, which overlays real-time and historical context—like flood risk, school quality, and air pollution levels—into property views; and **Automate**, where the agent automatically pulls data from multiple sources and updates insights with minimal input.

1. Local Context Analytics Interface

When users view a property, the system automatically presents key neighborhood metrics, including:

- Flood risk history and air quality trends
- Distance and quality scores for nearby schools, hospitals, and transit points

- Local news sentiment

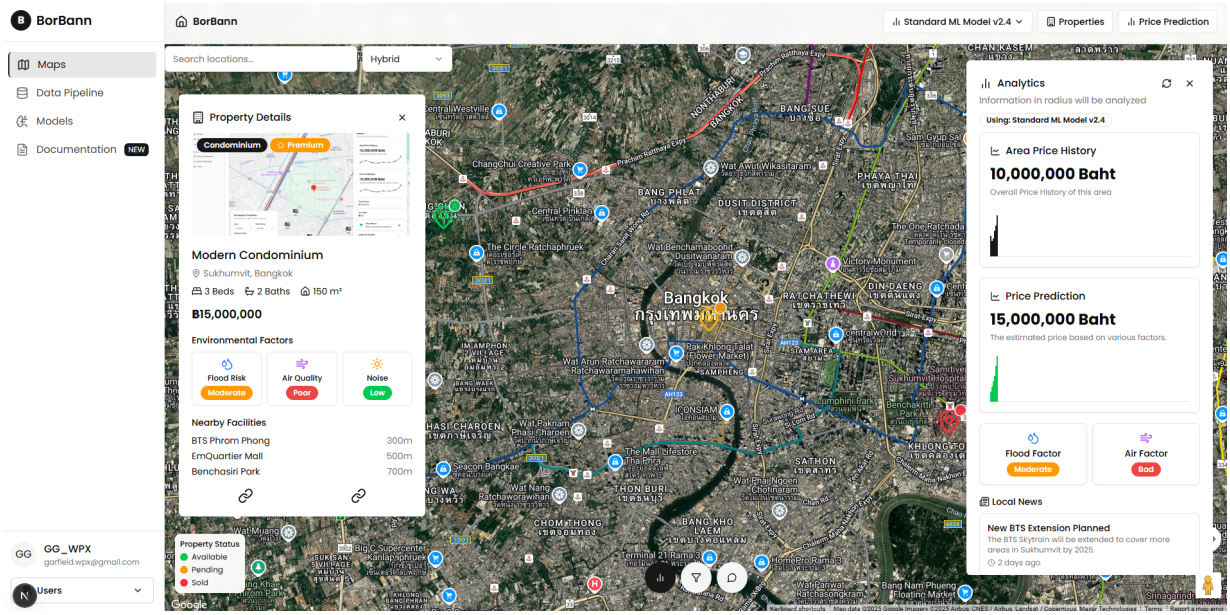

Figure 5.8: Environmental Impact Analysis

Figure 5.8 shows the analytics interface consisting of environmental data, pricing, and property-specific details along with nearby facilities.

Explainable Price Prediction

This feature blends predictive modeling with real-time visualizations and natural language explanations. Unlike typical black-box systems, BorBann's model lets users interact with the logic behind the number. Users can explore, adjust, and challenge the AI's assumptions.

The user experience combines:

- **Annotate** – each prediction includes visual and textual breakdowns of contributing factors.
- **Prompt** – the system suggests related factors for simulating "what-if" scenarios.
- **Automate** – the system recalibrates predictions as more data is integrated.

1. Prediction Overview

When a user selects a property, the system immediately displays:

- Predicted price range (upper and lower bounds)
- Confidence interval

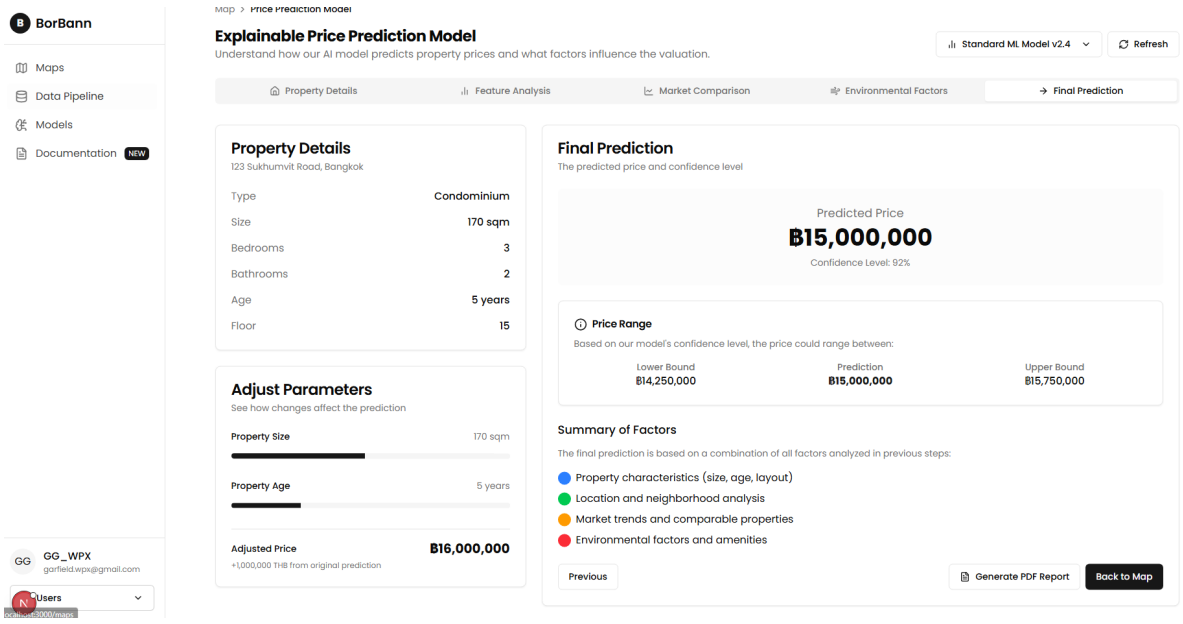

Figure 5.9: Prediction Overview Interface

- Base prediction with timestamp of latest data sync

Figure 5.9 shows the price range, confidence bands, and a clean summary layout.

2. Feature Contribution Analysis

The system explains which factors influenced the prediction most — such as location, property size, developer reputation, and local context.

Figure 5.10 shows an interface that visualizes the impact of each factor using bar graphs and percentage values.

5.5 Deployment Strategy

Objective

To detail the plan for integrating and running the AI-powered Real Estate Data Mapping model within a production environment, ensuring it effectively connects with the existing data ingestion pipeline and provides reliable, scalable, and maintainable operation. The AI model is central to transforming heterogeneous source data (from APIs, files, and web scraping) into a unified canonical format for real estate listings.

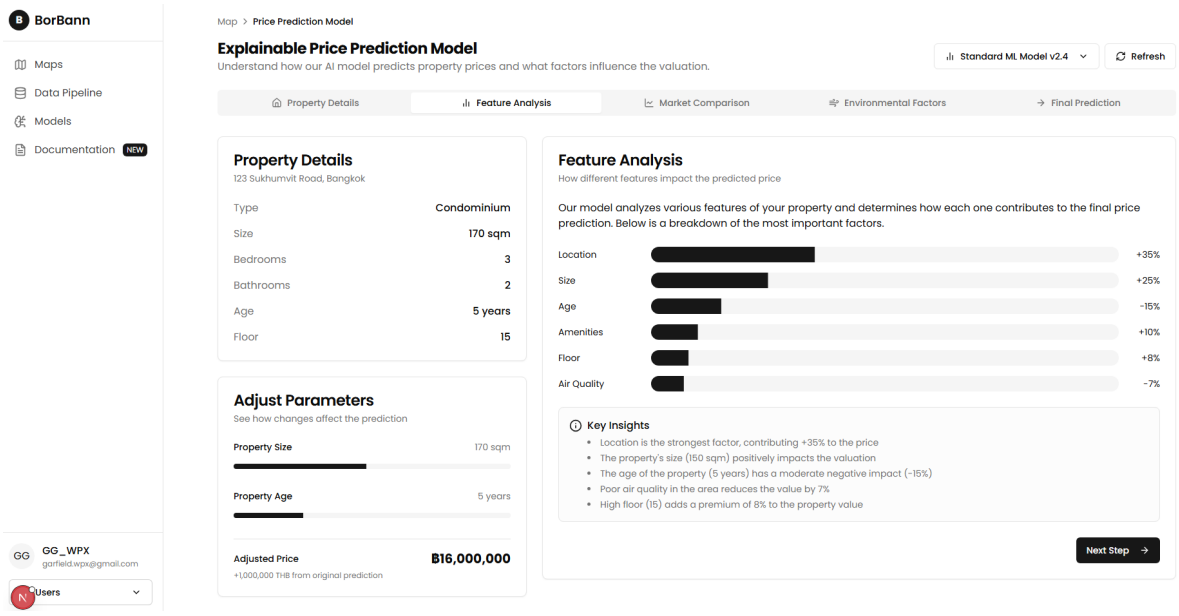

Figure 5.10: Feature Contribution Analysis Interface

Deployment Plan

Chosen Environment: Cloud-Based Deployment

The AI model, along with the entire data integration pipeline system, will be deployed on a **Cloud Platform** which is Google Cloud Platform (GCP). Vary between each AI component, for example, data schema mapping will use Vertex AI while other may use cloud computing.

Justification:

- **Scalability:** Cloud platforms offer elastic scaling of compute resources (CPUs, GPUs for LLM inference) and managed services, crucial for handling variable data loads and complex model computations.
- **Managed Services:** Leveraging services like Kubernetes (GKE, EKS, AKS) for container orchestration, managed databases for storing pipeline configurations and results, object storage (GCS, S3) for raw data and model artifacts, and potentially managed AI platforms (Vertex AI, SageMaker) simplifies infrastructure management.
- **Reliability & Availability:** Cloud providers offer high availability zones and built-in redundancy features.
- **Integration:** Easier integration with other cloud services (e.g., data warehouses, monitoring tools).

While on-device or embedded systems are not suitable for this large-scale data processing and LLM inference task, an Edge deployment could be considered in the future for specific data pre-processing or localized data collection components if required, but the core AI mapping will reside in the cloud.

AI Communication with the System: Internal Service Integration

The AI mapping model will not be exposed as a standalone, public-facing API initially. Instead, it will be integrated as an internal component within the existing data integration pipeline's backend service (built with FastAPI).

Communication Flow:

1. The `PipelineService` in the FastAPI application receives a request to run a specific pipeline (e.g., via its existing API endpoint `/pipelines/{pipeline_id}/run`).
2. If the pipeline configuration specifies the `ML_MAPPING` strategy, the `PipelineService` invokes the `IngestionStrategyFactory`.
3. The factory instantiates the `MLIngestionStrategy` (or a specialized version like `VertexAIMappingStrategy`).
4. The `MLIngestionStrategy` internally:
    - Loads or accesses the pre-trained/fine-tuned classification model (to identify real estate listings).
    - Loads or accesses the pre-trained/fine-tuned LLM mapping model (e.g., from an MLflow registry, a local path within the container, or by calling a managed AI service like Vertex AI).
    - Processes the input `AdapterRecord` data.
    - Returns the mapped `CanonicalRecord` objects (as `OutputData`) back to the `PipelineService`.
5. The `PipelineService` then stores these results.

This internal integration ensures that the AI model is a processing step within a larger workflow, rather than a standalone service that other parts of the system call directly via network requests for each mapping task. If a dedicated, reusable mapping service is needed later, a gRPC or REST API wrapper around the core mapping logic could be developed.

Tools and Frameworks Used

- **FastAPI:** For the main backend service orchestrating pipelines and exposing management APIs.
- **Python:** Primary programming language for the backend and AI model implementation.
- **Google Cloud Vertex AI** For using pre-trained foundation models (e.g., Gemini), fine-tuning models via Generative AI Studio, and deploying them as managed endpoints. Communication would be via the Google Cloud Python client libraries.
- **Pydantic:** For data validation of input, intermediate, and canonical schemas.

- **Loguru:** For structured logging throughout the application and AI components.
- **Cloud Storage (GCS, S3):** For storing raw input data, training datasets, and large model artifacts.
- **SQLite** For storing pipeline configurations, run metadata, and potentially links to canonical data results.

System Characteristics

Reliability:

- **Error Handling & Retries:**
  - The `PipelineService` will implement robust error handling for ingestion strategy failures (including AI model errors).
  - For transient issues (e.g., temporary unavailability of a cloud AI service), retry mechanisms (e.g., using libraries like 'tenacity') will be implemented for external API calls made by the `MLIngestionStrategy`.
  - The scheduler (APScheduler) has misfire grace time configurations.
- **Input Validation:** Pydantic validation at multiple stages (API input, canonical output) ensures data integrity.

Security:

- **Cloud IAM:** Utilize cloud provider Identity and Access Management (IAM) for granular control over access to resources (databases, storage, AI services, Kubernetes cluster). Principle of least privilege will be applied.
- **Secrets Management:** API keys for external LLMs (if used), database credentials, and other secrets will be managed using a secure secrets manager (e.g., HashiCorp Vault, Google Secret Manager, AWS Secrets Manager) and injected into containers as environment variables or mounted volumes, not hardcoded.
- **Data Encryption:** Data at rest (Cloud Storage, Databases) and in transit (HTTPS for external APIs, internal VPC traffic) will be encrypted.

Maintainability & Scalability:

- **Modular Design:** The separation of concerns (`PipelineService`, `IngestionStrategyFactory`, specific `Strategy` classes) allows for independent updates and maintenance of components.
- **Comprehensive Logging (Loguru):** Structured logs are centralized (e.g., Google Cloud Logging, ELK stack) for easier debugging and monitoring. The SSE log streaming endpoint aids real-time monitoring of specific pipeline runs.

- **Scalability (Application):** Kubernetes Horizontal Pod Autoscaler (HPA) will automatically scale the number of FastAPI application pods based on CPU/memory utilization or custom metrics.
- **Scalability (AI Model Inference):**
  - **Managed AI Service (e.g., Vertex AI):** These services typically handle autoscaling of the model endpoint based on traffic. Configure appropriate instance types and min/max replica counts.

Proof of Concept

AI Model Build and Test Process

The AI model for mapping real estate data to the `CanonicalRecord` schema was developed iteratively, focusing initially on an LLM-based approach.

Development Stages

1. **Data Collection & Annotation**

   A diverse dataset of approximately 500 property records was collected from various sources, including:

   - Simulated API outputs (e.g., mock property APIs)
   - Example CSV/JSON datasets
   - Real estate websites such as Baania and Zillow (conceptual scraping)

   These records were manually mapped to a unified schema called `CanonicalRecord`. A foundation model (e.g., GPT-4) was used to generate initial prompt-completion pairs via meta-prompting. All completions were manually reviewed and corrected to produce high-quality training data.

2. **LLM Mapper Fine-Tuning (Primary Task)**

   - **Model Selection:** Experiments were conducted on gemini-2.0-flash-lite-001.
   - **Fine-Tuning Strategy:** Fine-Tuning with supervised method on Vertex AI platform.
   - **Training Data:** About 30 prompt-completion pairs were used with 10 data point for evaluation set. Each prompt contained instructions, the full schema, and raw data. Each completion was a valid JSON object adhering to the `CanonicalRecord` structure.
   - **Platform:** All tuning jobs were executed on Vertex AI.
     - Hyperparameters: base model, learning rate, batch size, LoRA rank, LoRA alpha, and epoch count.
     - Outputs: Adapter weights, tokenizer config, and example outputs on the eval set.

3. **Evaluation Methodology**

   - **During Fine-Tuning:** Vertex AI provides training and validation losses automatically when an eval set is supplied.
   - **Post Fine-Tuning:**
     - **JSON Syntactic Validity:** Parse output strings using `json.loads()`.
     - **Pydantic Schema Conformance:** Instantiate `CanonicalRecord(**parsed_json)` to verify structural correctness.
   - **Manual Review:** A qualitative check was performed to ensure logical accuracy and edge case handling in LLM outputs.

Model Performance Results

Performance metrics are based on 2 metrics:

- **JSON Syntactic Validity**: Parse the output string and check for validity.
- **Pydantic Schema Conformance**: Check with pre-defined pydantic schema to ensure that it output the desire data scheme.

Table 5.1: Pipeline Validation Metrics

Model Version	Metric	Value (%)
BORBANN_PIPELINE_2	JSON Syntactic Validity	91.67%
	Pydantic Schema Conformance	63.64%
BORBANN_PIPELINE_3	JSON Syntactic Validity	100.00%
	Pydantic Schema Conformance	0.00%
BORBANN_PIPELINE_4	JSON Syntactic Validity	100.00%
	Pydantic Schema Conformance	0.00%

Table 5.1 presents the validation performance of three pipeline variants. Among them, `borbann-pipeline-2` achieves the highest score in JSON syntactic validity (91.67%), indicating that it produces well-formed JSON outputs most consistently. However, it performs best in this metric while showing only moderate conformance to the Pydantic schema (63.64%).

In contrast, both `borbann-pipeline-3` and `borbann-pipeline-4` attain perfect JSON syntactic validity (100%) but fail completely in Pydantic schema conformance (0.00%). This suggests that although their outputs are syntactically correct, they do not adhere to the expected canonical data structure.

Based on this evaluation, we select `borbann-pipeline-2` as the final model for deployment. The superior schema adherence—despite not being perfect—makes it more suitable for downstream structured processing tasks.

A possible reason for the low schema conformance in pipelines 3 and 4 may be sub-optimal prompt design during fine-tuning. The model may have overfit to an incorrect or inconsistent output structure due to insufficient coverage of schema variations in the training data. This highlights the importance of prompt engineering and data diversity when fine-tuning large language models for structured output tasks.

References

[1] J. Research, "Global real estate perspective february 2025: A positive but nuanced outlook for 2025," *JLL Global Real Estate Perspective*, February 2025. [Online]. Available: https://www.jll.co.th/en/trends-and-insights/research/global/gmp

[2] Nationthailand, "8 key trends shaping thai real estate after rollercoaster year," *Nationthailand*, January 2025. [Online]. Available: https://www.nationthailand.com/business/property/40044774

[3] B. Post, "Real estate information center targets better data gathering to head off crises," *Bangkok Post*, Jule 2024. [Online]. Available: https://www.bangkokpost.com/property/2836667/real-estate-information-center-targets-better-data-gathering-to-head-off-crises

[4] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. ACM, Aug. 2016, p. 785–794. [Online]. Available: http://dx.doi.org/10.1145/2939672.2939785

[5] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2018. [Online]. Available: https://arxiv.org/abs/1710.10903

[6] H. Lee, H. Jeong, B. Lee, K. Lee, and J. Choo, "St-rap: A spatio-temporal framework for real estate appraisal," 2023. [Online]. Available: https://arxiv.org/abs/2308.10609

[7] M. Mayer and D. Watson, *kernelshap: Kernel SHAP*, 2024, r package version 0.7.1. [Online]. Available: https://github.com/ModelOriented/kernelshap