TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

KỲ THI : OLYMPIC TIN HỌC 2025

HẠNG MỤC: PHẦN MỀM NGUỒN MỞ

HaUI | SCHOOL OF INFORMATION & COMMUNICATIONS TECHNOLOGY

CHỦ ĐỀ : "ỨNG DỤNG DỮ LIỆU MỞ LIÊN KẾT PHỤC VỤ CHUYỂN ĐỔI SỐ ĐỊA PHƯƠNG"

ĐỀ TÀI: "XÂY DỰNG ỨNG DỤNG NGUỒN MỞ TRỰC QUAN HÓA DỮ LIỆU CHUYỂN ĐỔI SỐ CẤP TỈNH VÀ MÔ HÌNH DỰ ĐOÁN HIỆU QUẢ DỊCH VỤ CÔNG TRỰC TUYẾN TỪ DỮ LIỆU CÔNG KHAI (OPEN DATA) CỦA BỘ TT&TT(2022-2024)"

GVHD : Nguyễn Thái Cường

Nhóm : HaUI.DNK

Người thực hiện :

Phạm Qúy Nam	2022606001
Trịnh Gia Luật	2022606228
Ngô Văn Tấn	2022606107

Hà Nội – 2025

MỤC LỤC

DANH MỤC THUẬT NGỮ VÀ VIẾT TẮT

Ký hiệu/ Viết tắt	Tiếng Anh đầy đủ (Full Term)	Nghĩa tiếng Việt (Meaning)
CĐS		Chuyển đổi số
DTI	Digital Transformation Index	Bộ chỉ số đánh giá Chuyển đổi số
GSO	General Statistics Office	Các số liệu, thông tin, hoặc báo cáo về tình hình Kinh tế - Xã hội *theo nguồn từ Tổng cục Thống kê*
DVC		Dịch vụ công
HTML	HyperText Markup Language	Ngôn ngữ Đánh dấu Siêu văn bản
PDF	Portable Document Format	Định dạng Tài liệu Di động
API	Application Programming Interface	Giao diện Lập trình Ứng dụng
ML	Machine Learning	Học máy
ETL Pipeline	Extract, Transform, Load Pipeline	Đường ống ETL
GDP	Gross Domestic Product	Tổng sản phẩm quốc nội
R2	R-squared	Chỉ số đo lường hiệu suất
MSE	Mean Squared Error	Lỗi Bình phương Trung bình
Bộ TT&TT		Bộ Thông tin và Truyền thông

DANH MỤC HÌNH ẢNH

MỞ ĐẦU

1.Lý do chọn đề tài

Chuyển đổi số (CĐS) là xu hướng tất yếu và là một trong những ưu tiên hàng đầu của Chính phủ Việt Nam. Để thúc đẩy quá trình này, nhiều bộ ngành và địa phương đã bắt đầu công khai dữ liệu mở trên các cổng thông tin quốc gia (data.gov.vn, opendata.mic.gov.vn) và các cổng dữ liệu địa phương.

Tuy nhiên, việc khai thác giá trị thực sự từ các nguồn này đang gặp ba thách thức lớn:

- Tính phân tán: Dữ liệu về CĐS (DTI), Kinh tế-Xã hội (GSO), và Dịch vụ công (DVC) nằm ở nhiều nguồn, nhiều định dạng (bảng HTML, PDF, API) khác nhau, gây khó khăn cho việc tổng hợp và phân tích đồng bộ.
- Thiếu trực quan hóa: Cộng đồng, doanh nghiệp và các nhà quản lý thiếu một công cụ tập trung để theo dõi, so sánh thứ hạng và đánh giá hiệu quả CĐS giữa các địa phương một cách trực quan.
- Thiếu khai phá tri thức: Dữ liệu thô chưa được khai thác bằng các mô hình Học máy (ML) để tìm ra các quy luật ngầm và trả lời câu hỏi cốt lõi: "Trong các yếu tố (Hạ tầng, Kinh tế, Dân số, Dịch vụ công), yếu tố nào có tác động mạnh mẽ nhất đến chỉ số DTI của một tỉnh?"

Xuất phát từ những thách thức thực tiễn đó, đề tài "Xây dựng ứng dụng mã nguồn mở trực quan hóa và dự đoán hiệu quả Chuyển đổi số cấp tỉnh" được lựa chọn để giải quyết các vấn đề trên.

2.Mục tiêu nghiên cứu

Mục tiêu tổng quát của dự án là xây dựng một hệ thống ứng dụng web mã nguồn mở có khả năng tích hợp, trực quan hóa và dự đoán các chỉ số CĐS từ dữ liệu công khai.

Các mục tiêu cụ thể bao gồm:

- Xây dựng một quy trình Tiền xử lý Dữ liệu (ETL Pipeline) tự động để tổng hợp và hợp nhất 3 nguồn dữ liệu (DTI, GSO, DVC) thành một bộ dữ liệu sạch, duy nhất.
- Xây dựng mô hình Học máy (cụ thể là Random Forest) để dự đoán chỉ số DTI_Tong và định lượng mức độ quan trọng (Feature Importance) của 9 đặc trưng đầu vào.
- Xây dựng một ứng dụng Dashboard (React) cung cấp 3 chức năng chính: (1) Trực quan hóa so sánh 34 tỉnh, (2) Phân tích chi tiết xu hướng theo từng tỉnh, và (3) Mô phỏng dự đoán "What-if".

3.Đối tượng và phạm vi nghiên cứu

Đối tượng nghiên cứu: Mối quan hệ tương quan và nhân quả giữa các yếu tố đầu vào (Hạ tầng số, Dân số, GDP, 5 chỉ số DVC) và chỉ số hiệu quả Chuyển đổi số (DTI_Tong).

Phạm vi nghiên cứu:

- *Về dữ liệu:* Giới hạn trong 3 nguồn dữ liệu (Bảng 1 - DTI, Bảng 2 - GSO, Bảng 3 - DVC).
- *Về không gian (Tỉnh):* Giới hạn trong 34 tỉnh/thành phố có đầy đủ dữ liệu từ cả 3 nguồn.
- *Về thời gian:* Giới hạn trong giai đoạn 2022 - 2024.

(Bộ dữ liệu cuối cùng bao gồm 102 mẫu (34 tỉnh x 3 năm)).

4.Phương pháp nghiên cứu

Dự án sử dụng kết hợp nhiều phương pháp khoa học dữ liệu và kỹ thuật phần mềm:

Phương pháp Thu thập Dữ liệu:

- *Web Scraping (HTML):* Dùng pandas.read_html để trích xuất dữ liệu DTI.
- *PDF Table Extraction:* Dùng camelot-py để trích xuất dữ liệu GSO từ Niên giám PDF.

- *API Sniffing & Calling:* Dùng requests.post để gọi API ẩn, lấy dữ liệu DVC.

Phương pháp Xử lý Dữ liệu:

- *Thống kê mô tả và Tiền xử lý:* Dùng pandas để làm sạch, chuẩn hóa tên tỉnh (Regex), và tạo đặc trưng mới (TyLeThanhThi).
- *Hợp nhất dữ liệu:* Sử dụng INNER JOIN để tạo bộ dữ liệu sạch.

Phương pháp Mô hình hóa (Học máy):

- *Chuẩn hóa:* Sử dụng StandardScaler của Scikit-learn.
- *Tối ưu hóa:* Sử dụng GridSearchCV để tìm siêu tham số tốt nhất cho mô hình RandomForestRegressor.
- *Đánh giá:* Sử dụng các chỉ số R-squared (R2) và MSE để đo lường hiệu quả mô hình.

Phương pháp Phát triển Phần mềm:

- *Kiến trúc:* 3-Lớp (React Frontend, Flask Backend, Data Layer).
- *Triển khai:* Đóng gói ứng dụng Backend bằng Docker.

5.Ý nghĩa khoa học và thực tiễn

Ý nghĩa khoa học:

Đề xuất một quy trình (pipeline) hoàn chỉnh và khả thi để tổng hợp 3 nguồn dữ liệu CĐS vốn bị phân tán và không đồng nhất.

Định lượng hóa một cách khách quan mức độ ảnh hưởng của các yếu tố đến Chuyển đổi số. Kết quả nghiên cứu (chỉ ra HaTangSo là yếu tố quan trọng nhất) là một phát hiện có giá trị, cung cấp bằng chứng dựa trên dữ liệu.

Ý nghĩa thực tiễn:

Cung cấp một **công cụ Dashboard trực quan, mã nguồn mở** cho cộng đồng, doanh nghiệp và các nhà quản lý để theo dõi và so sánh hiệu quả CĐS (điều mà trước đây thiếu).

Cung cấp **công cụ Mô phỏng Dự đoán ("What-if")** có tính ứng dụng cao, giúp hỗ trợ ra quyết định chiến lược (ví dụ: "Tỉnh X nên ưu tiên đầu tư vào Hạ tầng hay Dịch vụ công để tăng DTI hiệu quả nhất?").

CHƯƠNG 1: GIỚI THIỆU TỔNG QUAN

1.1.Bối cảnh và Bài toán

Trong tiến trình chuyển đổi số (CĐS) cấp tỉnh, nhiều địa phương tại Việt Nam đã công khai dữ liệu mở trên các cổng thông tin điện tử như data.gov.vn (Cổng dữ liệu quốc gia), opendata.mic.gov.vn (Bộ TT&TT), và các cổng dữ liệu địa phương (opendata.danang.gov.vn, opendata.hochiminhcity.gov.vn...).

Mặc dù dữ liệu đã được công khai, việc khai thác giá trị thực sự từ các nguồn này đang gặp một thách thức lớn, bao gồm 3 vấn đề cốt lõi:

Vấn đề Phân tán & Thiếu đồng bộ:

Các nguồn dữ liệu CĐS (DTI), Kinh tế (GSO), Dịch vụ công (DVC) nằm rời rạc, không nhất quán về cấu trúc và định dạng (web, PDF, API). Điều này khiến việc khai thác đồng bộ trở nên rất khó khăn.

Vấn đề Thiếu Trực quan hóa:

Người dân, doanh nghiệp và nhà quản lý thiếu một công cụ (dashboard) tập trung để theo dõi xu hướng, so sánh xếp hạng, và đánh giá hiệu quả CĐS một cách trực quan, nhanh chóng.

Vấn đề Thiếu Khai phá Tri thức:

Dữ liệu thô chưa được khai thác bằng Học máy (ML) để trả lời các câu hỏi chiến lược, ví dụ: "Yếu tố nào (Hạ tầng số, Kinh tế, hay DVC) tác động mạnh nhất đến Chuyển đổi số?"

1.2.Giải pháp và Mục tiêu

Để giải quyết bài toán trên, dự án xây dựng một Hệ thống Web Mã nguồn mở Toàn diện với 3 mục tiêu chính:

*** TỔNG HỢP (Consolidate):**

Xây dựng một quy trình (pipeline) tự động để thu thập, làm sạch và hợp nhất (join) 3 nguồn dữ liệu rời rạc thành một bộ dữ liệu (master_df) duy nhất.

*** TRỰC QUAN HÓA (Visualize):**

Cung cấp một Dashboard tương tác (React) để so sánh 34 tỉnh/thành phố trên nhiều khía cạnh (DTI, GDP, DVC) và theo dõi xu hướng qua 3 năm (2022-2024).

*** DỰ ĐOÁN (Predict):**

Xây dựng mô hình Machine Learning (Random Forest) để tìm ra các yếu tố ảnh hưởng mạnh nhất đến DTI.

Cung cấp công cụ mô phỏng "What-if" để dự đoán DTI dựa trên các kịch bản đầu vào.

CHƯƠNG 2: KIẾN TRÚC HỆ THỐNG

2.1. Cấu trúc Hệ thống Tổng thể

Cấu trúc tổng thể của dự án được thiết kế theo một luồng xử lý dữ liệu (data pipeline) hoàn chỉnh, từ dữ liệu thô đến sản phẩm ứng dụng.

- **Khối 1: Thu thập Dữ liệu (Data Ingestion)**

  Thu thập 3 nguồn dữ liệu thô (DTI, GSO, DVC) dưới dạng file CSV/PDF hoặc gọi API.

- **Khối 2: Tiền xử lý Dữ liệu (Data Preprocessing)**

  Sử dụng kịch bản Python (Pandas) để chuẩn hóa tên tỉnh, tạo đặc trưng mới, và thực hiện INNER JOIN để tạo ra tệp dữ liệu sạch master_data_cleaned_merged.csv (102 dòng).

- **Khối 3: Huấn luyện Mô hình (Model Training)**

  Sử dụng kịch bản Python (Scikit-learn) trên tệp dữ liệu sạch.

  Quy trình này thực hiện StandardScaler, GridSearchCV (để tối ưu Random Forest) và xuất ra 2 tệp: random_forest_model.joblib và features_scaler.joblib.

- **Khối 4: Backend (API Server)**

  Một API server (Flask/Gunicorn) được "Docker hóa", có nhiệm vụ tải 3 tệp (.csv, .joblib, .joblib) vào bộ nhớ để cung cấp API cho Frontend.

- **Khối 5: Frontend (Client Application)**

  Một ứng dụng Dashboard (React) chạy trên trình duyệt, gọi API, nhận dữ liệu JSON và trực quan hóa cho người dùng.

2.2. Kiến trúc Phần mềm 3-Lớp (3-Tier Architecture)

Ứng dụng web được thiết kế theo kiến trúc 3-Lớp (3-Tier) tiêu chuẩn:

2.2.1 Tầng Trình diễn (Presentation Layer - Frontend)

Công nghệ: React.js, Chart.js (Biểu đồ), Leaflet.js (Bản đồ).

Nhiệm vụ: Hiển thị 3 giao diện chức năng (Tổng quan, Chi tiết, Dự đoán) và quản lý tương tác người dùng. Tầng này chạy hoàn toàn trên trình duyệt của client.

2.2.2 Tầng Logic/Ứng dụng (Business Logic Layer - Backend)

Công nghệ: Flask, Gunicorn, Docker.

Nhiệm vụ: Cung cấp logic nghiệp vụ qua 2 API endpoint (/api/data, /api/predict). Đây là nơi chứa "bộ não" của hệ thống (mô hình ML).

2.2.3. Tầng Dữ liệu (Data Layer)

Công nghệ: Hệ thống file (File System) trong Docker container.

Nhiệm vụ: Lưu trữ 3 tệp tài nguyên tĩnh:

- master_data_cleaned_merged.csv (Cơ sở dữ liệu cho Dashboard)
- random_forest_model.joblib (Mô hình ML)
- features_scaler.joblib (Bộ chuẩn hóa)

CHƯƠNG 3: QUY TRÌNH XỬ LÝ DỮ LIỆU (ETL PIPELINE)

3.1. Nguồn Dữ liệu

Dự án tích hợp 3 nguồn dữ liệu độc lập:

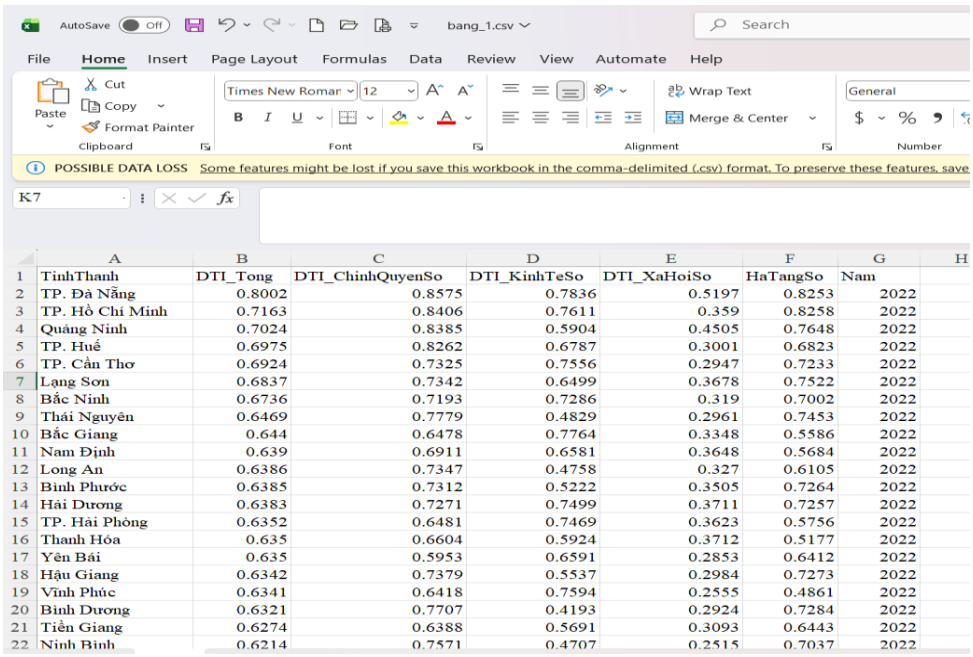
-Bảng 1 (DTI - Bộ TT&TT):

Nguồn: dti.gov.vn

Kỹ thuật: Web Scraping (cào bảng HTML).

Dữ liệu chính: DTI_Tong, DTI_ChinhQuyenSo, DTI_KinhTeSo, DTI_XaHoiSo, HaTangSo.

Kích thước: 189 dòng (63 tỉnh x 3 năm).



	A	B	C	D	E	F	G	H
1	TinhThanh	DTI_Tong	DTI_ChinhQuyenSo	DTI_KinhTeSo	DTI_XaHoiSo	HaTangSo	Nam	
2	TP. Đà Nẵng	0.8002	0.8575	0.7836	0.5197	0.8253	2022	
3	TP. Hồ Chí Minh	0.7163	0.8406	0.7611	0.359	0.8258	2022	
4	Quảng Ninh	0.7024	0.8385	0.5904	0.4505	0.7648	2022	
5	TP. Huế	0.6975	0.8262	0.6787	0.3001	0.6823	2022	
6	TP. Cần Thơ	0.6924	0.7325	0.7556	0.2947	0.7233	2022	
7	Lạng Sơn	0.6837	0.7342	0.6499	0.3678	0.7522	2022	
8	Bắc Ninh	0.6736	0.7193	0.7286	0.319	0.7002	2022	
9	Thái Nguyên	0.6469	0.7779	0.4829	0.2961	0.7453	2022	
10	Bắc Giang	0.644	0.6478	0.7764	0.3348	0.5586	2022	
11	Nam Định	0.639	0.6911	0.6581	0.3648	0.5684	2022	
12	Long An	0.6386	0.7347	0.4758	0.327	0.6105	2022	
13	Bình Phước	0.6385	0.7312	0.5222	0.3505	0.7264	2022	
14	Hải Dương	0.6383	0.7271	0.7499	0.3711	0.7257	2022	
15	TP. Hải Phòng	0.6352	0.6481	0.7469	0.3623	0.5756	2022	
16	Thanh Hóa	0.635	0.6604	0.5924	0.3712	0.5177	2022	
17	Yên Bái	0.635	0.5953	0.6591	0.2853	0.6412	2022	
18	Hậu Giang	0.6342	0.7379	0.5537	0.2984	0.7273	2022	
19	Vĩnh Phúc	0.6341	0.6418	0.7594	0.2555	0.4861	2022	
20	Bình Dương	0.6321	0.7707	0.4193	0.2924	0.7284	2022	
21	Tiền Giang	0.6274	0.6388	0.5691	0.3093	0.6443	2022	
22	Ninh Bình	0.6214	0.7571	0.4707	0.2515	0.7037	2022	

Hình 3.1.1 Dữ liệu DTI

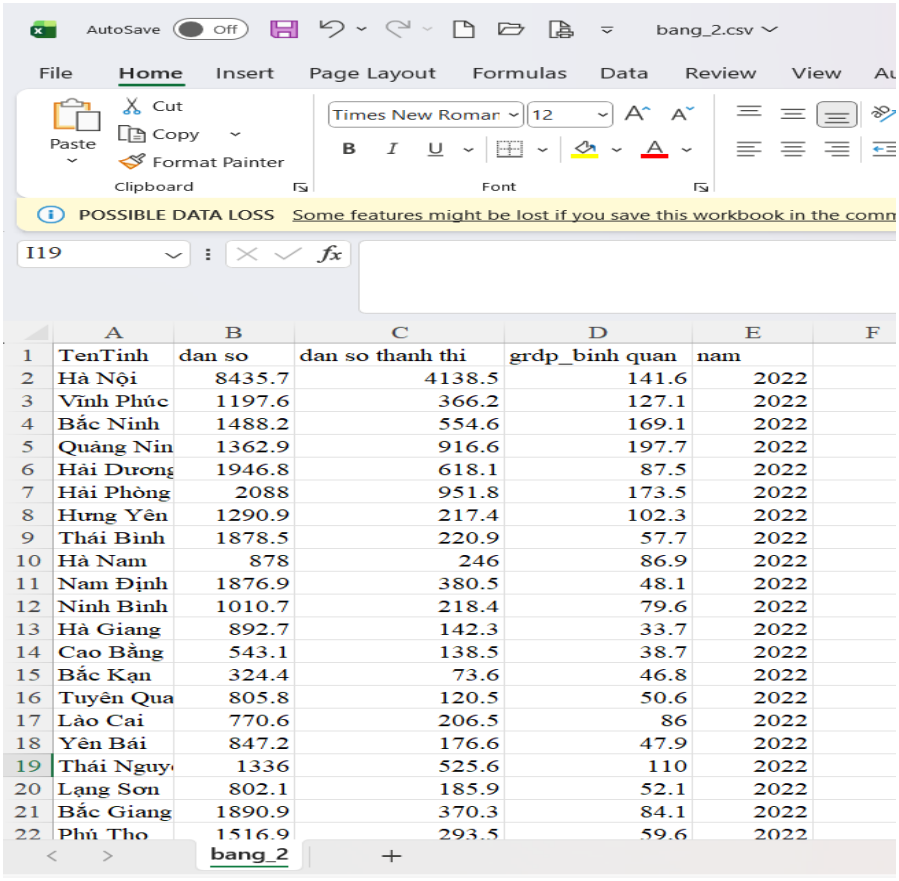
-Bảng 2 (Kinh tế-Xã hội - GSO):

Nguồn: Niên giám Thống kê (PDF).

Kỹ thuật: PDF Table Extraction (dùng camelot-py).

Dữ liệu chính: dan so, dan so thanh thi, grdp_binh quan.

Kích thước: 189 dòng (63 tỉnh x 3 năm).



	A	B	C	D	E	F
1	TenTinh	dan so	dan so thanh thi	grdp_binh quan	nam	
2	Hà Nội	8435.7	4138.5	141.6	2022	
3	Vĩnh Phúc	1197.6	366.2	127.1	2022	
4	Bắc Ninh	1488.2	554.6	169.1	2022	
5	Quảng Nin	1362.9	916.6	197.7	2022	
6	Hải Dương	1946.8	618.1	87.5	2022	
7	Hải Phòng	2088	951.8	173.5	2022	
8	Hưng Yên	1290.9	217.4	102.3	2022	
9	Thái Bình	1878.5	220.9	57.7	2022	
10	Hà Nam	878	246	86.9	2022	
11	Nam Định	1876.9	380.5	48.1	2022	
12	Ninh Bình	1010.7	218.4	79.6	2022	
13	Hà Giang	892.7	142.3	33.7	2022	
14	Cao Bằng	543.1	138.5	38.7	2022	
15	Bắc Kạn	324.4	73.6	46.8	2022	
16	Tuyên Qua	805.8	120.5	50.6	2022	
17	Lào Cai	770.6	206.5	86	2022	
18	Yên Bái	847.2	176.6	47.9	2022	
19	Thái Nguy	1336	525.6	110	2022	
20	Lạng Sơn	802.1	185.9	52.1	2022	
21	Bắc Giang	1890.9	370.3	84.1	2022	
22	Phú Thọ	1516.9	293.5	59.6	2022	

bang_2

Hình 3.1.2 Dữ liệu Kinh tế-xã hội

-Bảng 3 (Dịch vụ công - DVC):

Nguồn: API ẩn của dichvucong.gov.vn.

Kỹ thuật: API Sniffing (dùng requests.post với payload đã phân tích).

Dữ liệu chính: 5 chỉ số chi tiết (CongKhaiMinhBach, TienDoGiaiQuyet, DichVuTrucTuyen, MucDoHaiLong, SoHoaHoSo).

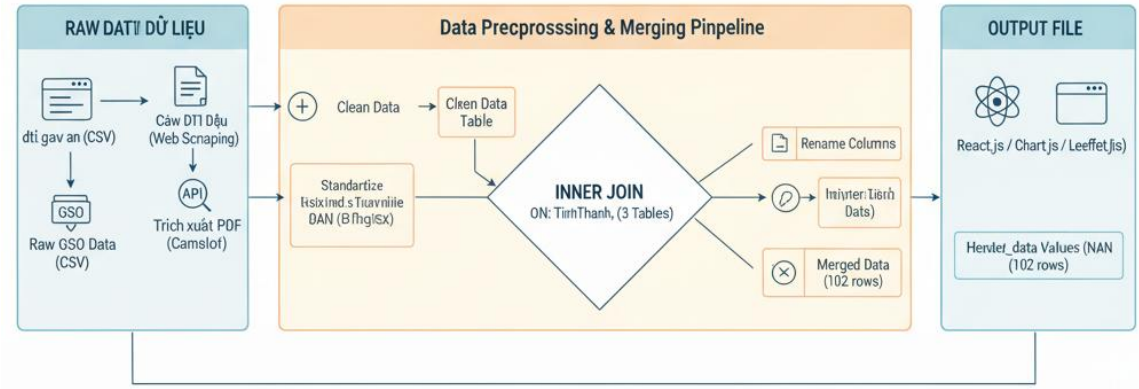
Kích thước: 102 dòng (Chỉ có 34 tỉnh x 3 năm).

	A	B	C	D	E	F	G	H	I
1	TinhThanh	TongDiem_DVC	CongKhaiMinhBach	TienDoGiaiQuyet	DichVuTrucTuyen	MucDoHaiLong	SoHoaHoSo	Nam	
2	UBND tỉnh Khánh Hòa	65.06	9.24	19.05	6.9	17.68	12.19	2022	
3	UBND tỉnh Ninh Bình	61.56	7.23	19.59	5.53	18	11.21	2022	
4	UBND Thành phố Hồ C	59.84	3.92	19.4	3.54	17.73	15.25	2022	
5	UBND tỉnh Lào Cai	59.83	11.28	19.48	5.05	17.46	6.56	2022	
6	UBND tỉnh Tây Ninh	59.21	7.47	19.27	5.16	17.7	9.61	2022	
7	UBND tỉnh Cà Mau	58.9	8.15	19.16	5.55	17.79	8.25	2022	
8	UBND tỉnh Thái Nguyê	58.85	8.45	18.65	5.66	17.88	8.21	2022	
9	UBND tỉnh Lai Châu	58.84	9.2	19.36	5.45	18	6.83	2022	
10	UBND tỉnh Lâm Đồng	58.56	7.56	18.85	7.13	17.94	7.08	2022	
11	UBND Thành phố Huế	58.43	9.34	16.49	5.79	16.97	9.84	2022	
12	UBND tỉnh Quảng Ninh	57.67	5.87	19.08	5.58	17.76	9.38	2022	
13	UBND tỉnh Hà Tĩnh	57.01	7.14	17.81	5.64	15.57	10.85	2022	
14	UBND Thành phố Đà N	56.32	7.71	17.3	5.82	16.3	9.19	2022	
15	UBND tỉnh Lạng Sơn	55.69	7.56	19.22	4.16	16.16	8.59	2022	
16	UBND tỉnh Tuyên Quan	55.46	5.65	18.71	4.92	18	8.18	2022	
17	UBND tỉnh Điện Biên	55.45	6.75	19.42	4.6	18	6.68	2022	
18	UBND tỉnh Đồng Nai	55.16	11.19	19.23	3.28	15.81	5.65	2022	
19	UBND tỉnh Quảng Ngãi	54.8	7.28	18.99	6.17	17.19	5.17	2022	
20	UBND tỉnh Phú Thọ	54.64	7.05	19.5	4.38	17.21	6.5	2022	
21	UBND tỉnh Hưng Yên	54.62	7.66	19.41	4.44	18	5.11	2022	
22	UBND tỉnh Thanh Hóa	54.41	9.79	16.36	4.72	16.21	7.33	2022	

bang_3

Hình 3.1.3 Dữ liệu Dịch vụ công

3.2. Sơ đồ Luồng Dữ liệu (Data Flow Diagram)



Hình 3.2.1 Sơ đồ Luồng Dữ liệu

3.3. Giai đoạn Biến đổi (Transform)

Giai đoạn Biến đổi (Transform) là lõi kỹ thuật của quy trình Tiền xử lý Dữ liệu, giải quyết hai vấn đề chính:

3.3.1 Kỹ thuật Đặc trưng (Feature Engineering) (Bảng 2)

Dữ liệu gốc từ GSO chỉ có dan so và dan so thanh thi.

Một đặc trưng mới, TyLeThanhThi, được tạo ra bằng công thức: df['TyLeThanhThi'] = (df['DanSoThanhThi'] / df['DanSo']) * 100

```
# --- Xử lý Bảng 2 (GSO) ---
df2.rename(columns={
    'nam': 'Nam', 'dan so': 'DanSo', 'dan so thanh thi': 'DanSoThanhThi',
    'grdp_binh quan': 'GRDP_BinhQuan', 'TenTinh': 'TinhThanh'
}, inplace=True)
df2['TyLeThanhThi'] = (df2['DanSoThanhThi'] / df2['DanSo']) * 100
df2 = df2.drop(columns=['DanSoThanhThi'])
```

Hình 3.2.2 Xử lý logic bảng GSO

Đặc trưng này sau đó được chứng minh là một trong những yếu tố quan trọng nhất trong mô hình dự đoán.

3.3.2. Chuẩn hóa Tên tỉnh (Thách thức lớn nhất)

Ba nguồn dữ liệu sử dụng định dạng tên tỉnh khác nhau:

- Bảng 1: "TP. Đà Nẵng", "Thừa Thiên Huế"
- Bảng 2: "Đà Nẵng", "Thừa Thiên - Huế"
- Bảng 3: "UBND Thành phố Đà Nẵng", "UBND tỉnh Thừa Thiên Huế"

Giải pháp: Xây dựng một hàm standardize_province_name sử dụng Biểu thức Chính quy (Regular Expression) để loại bỏ tiền tố (UBND..., Tỉnh...) và chuẩn hóa các trường hợp đặc biệt (ví dụ: Huế $\rightarrow$ Thừa Thiên Huế), đảm bảo 3 bảng có thể join được với nhau.

```python
def standardize_province_name(name):
    """
    Hàm trợ giúp:
    Làm sạch và chuẩn hóa tên tỉnh về một định dạng chung.
    """
    name = str(name).strip()
    # Bỏ các tiền tố chung
    name = re.sub(r'^(TP\. |Tỉnh |Thành phố )', '', name)
    # Chuẩn hóa các trường hợp đặc biệt
    if 'Hồ Chí Minh' in name:
        return 'TP. Hồ Chí Minh'
    if 'Huế' in name or 'Thừa Thiên' in name:
        return 'Thừa Thiên Huế'
    if 'Bà Rịa' in name:
        return 'Bà Rịa - Vũng Tàu'
    if 'Đắk Lắk' in name:
        return 'Đắk Lắk'
    if 'Đắk Nông' in name:
        return 'Đắk Nông'
    return name

print("Đã định nghĩa hàm 'standardize_province_name'.")
```

Hình 3.3.1 Xử lý logic trong hàm standardize_province_name

3.4. Giai đoạn Hợp nhất (Merge)

Vấn đề: Bảng 1 và 2 có đủ 63 tỉnh, nhưng Bảng 3 (DVC) chỉ có dữ liệu của 34 tỉnh.

Giải pháp: Sử dụng phương thức pandas.merge với how='inner' (Hợp nhất giao).

Logic: "Chỉ giữ lại những dòng (cặp Tỉnh-Năm) nào có mặt ở CẢ 3 BẢNG."

Kết quả: Bộ dữ liệu cuối cùng (master_df) bao gồm 102 dòng (34 tỉnh x 3 năm) với 16 cột đặc trưng hoàn chỉnh, không có dữ liệu khuyết.

```python
# --- Hợp nhất (INNER JOIN) ---
df_merged = pd.merge(df1, df2, on=['TinhThanh', 'Nam'], how='inner')
master_df = pd.merge(df_merged, df3, on=['TinhThanh', 'Nam'], how='inner')
```

Hình 3.4.1 Hợp nhất dữ liệu

Ý nghĩa: Quyết định này đảm bảo mô hình ML được huấn luyện trên dữ liệu chất lượng cao, đầy đủ, và Dashboard sẽ tập trung phân tích 34 tỉnh/thành phố trọng điểm này.

CHƯƠNG 4: XÂY DỰNG MÔ HÌNH DỰ ĐOÁN

4.1. Lựa chọn Đặc trưng (Feature Selection)

Dựa trên giả thuyết của dự án, chúng tôi đã chọn 9 đặc trưng đầu vào để dự đoán 1 biến mục tiêu.

Biến Mục tiêu (y):

DTI_Tong: Chỉ số Chuyển đổi số Tổng hợp (thể hiện "hiệu quả CĐS").

Bộ Đặc trưng (X) (9 cột):

- HaTangSo: Hạ tầng số (từ Bảng 1).
- DanSo: Dân số (từ Bảng 2).
- GRDP_BinhQuan: GDP bình quân (từ Bảng 2).
- TyLeThanhThi: Tỷ lệ đô thị hóa (từ Bảng 2).
- CongKhaiMinhBach: Chỉ số DVC (từ Bảng 3).
- TienDoGiaiQuyet: Chỉ số DVC (từ Bảng 3).
- DichVuTrucTuyen: Chỉ số DVC (từ Bảng 3).
- MucDoHaiLong: Chỉ số DVC (từ Bảng 3).
- SoHoaHoSo: Chỉ số DVC (từ Bảng 3).

```python
# Biến mục tiêu (y)
y = master_df['DTI_Tong']

# Biến đặc trưng (X)
feature_columns = [
    'HaTangSo',
    'DanSo',
    'GRDP_BinhQuan',
    'TyLeThanhThi',
    'CongKhaiMinhBach',
    'TienDoGiaiQuyet',
    'DichVuTrucTuyen',
    'MucDoHaiLong',
    'SoHoaHoSo'
]
X = master_df[feature_columns]
```

Hình 4.1.1 Cấu hình Biến Mục tiêu và Bộ Đặc trưng

4.2. Lựa chọn và Tối ưu hóa Mô hình (Model Selection & Optimization)

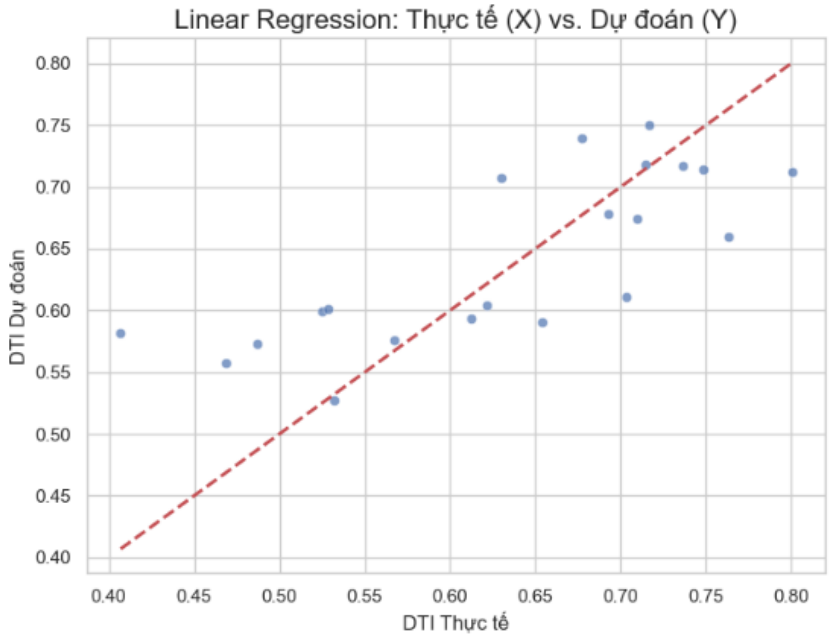
Một quy trình thử nghiệm đã được thực hiện để chọn ra mô hình tốt nhất:

Mô hình Cơ sở (Baseline):

LinearRegression được huấn luyện. Kết quả cho 56.8%. Đây là kết quả tốt, cho thấy các đặc trưng có mối liên hệ tuyến tính, nhưng bị giới hạn bởi giả định "đường thẳng".

```
--- KẾT QUẢ LINEAR REGRESSION ---
  R-squared (R2): 0.5681
  Mean Squared Error (MSE): 0.0049
```

Hình 4.2.1 Kết quả mô hình Linear regression



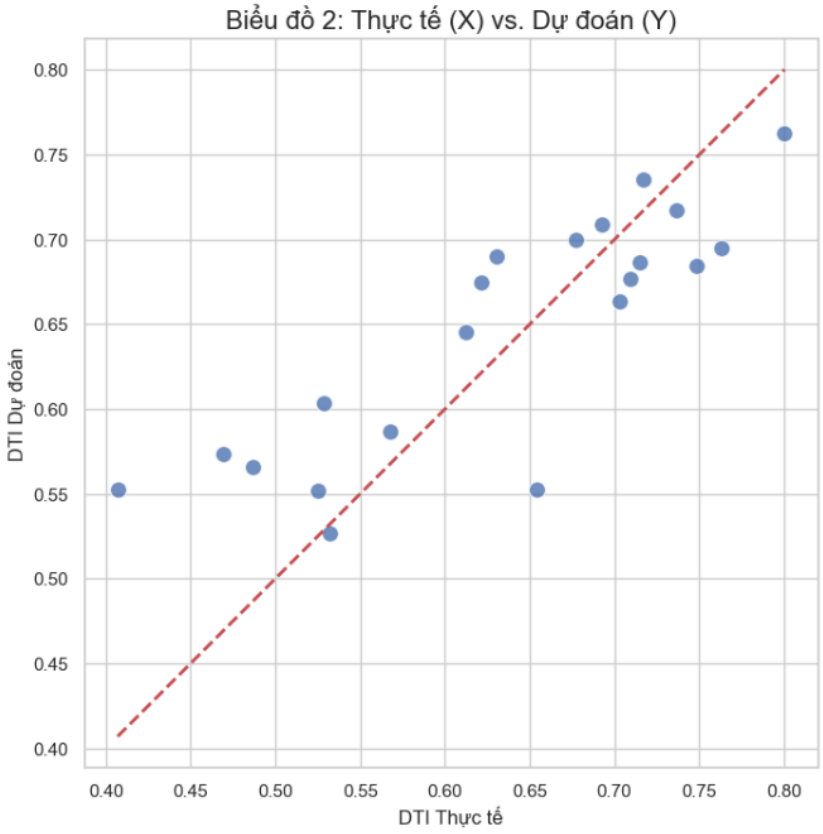
Hình 4.2.2 Kết quả mô hình Linear regression

Mô hình Tối ưu (Champion):

RandomForestRegressor được lựa chọn vì khả năng học các mối quan hệ phi tuyến (non-linear) và tương tác đặc trưng phức tạp (ví dụ: Hạ tầng chỉ hiệu quả khi Dân số đông).

```
--- KẾT QUẢ RANDOM FOREST TỐI ƯU ---
  R-squared (R2): 0.6707 (tức 67.07%)
  Mean Squared Error (MSE): 0.0037
```

Hình 4.2.3 Kết quả mô hình Random forest



Hình 4.2.4 Kết quả mô hình Random forest

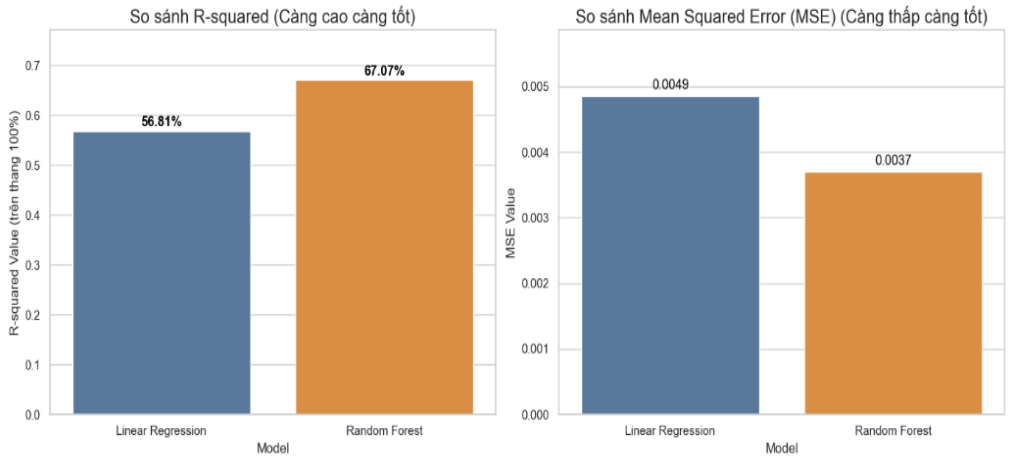
Tối ưu hóa (Optimization):

Chúng tôi sử dụng GridSearchCV (Dò tìm tham số lưới) của Scikit-learn.

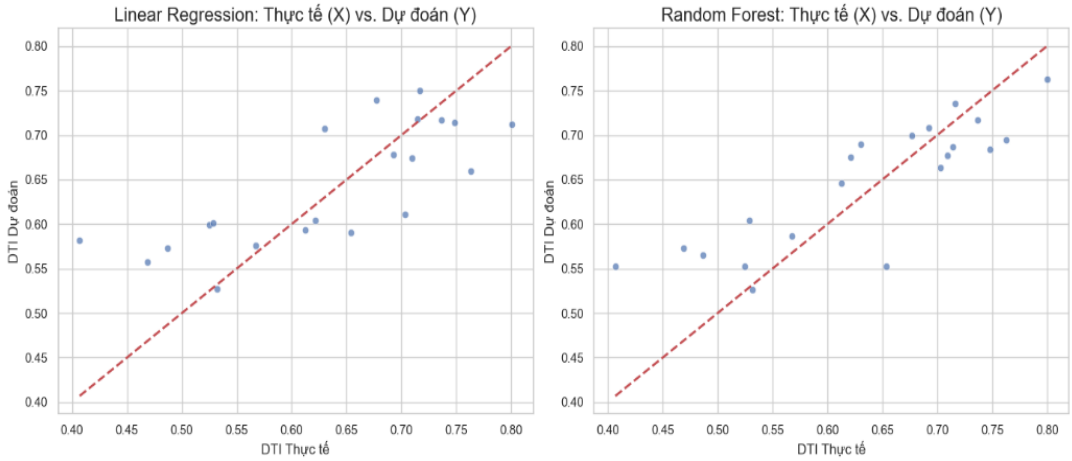
Quá trình này tự động thử nghiệm nhiều tổ hợp tham số (n_estimators, max_depth...) để tìm ra cấu hình Random Forest tốt nhất, tối ưu hóa theo chỉ số R^2.

4.3. Kết quả và Phân tích Nổi bật

Mô hình Random Forest tối ưu cho kết quả R^2 vượt trội so với mô hình cơ sở.

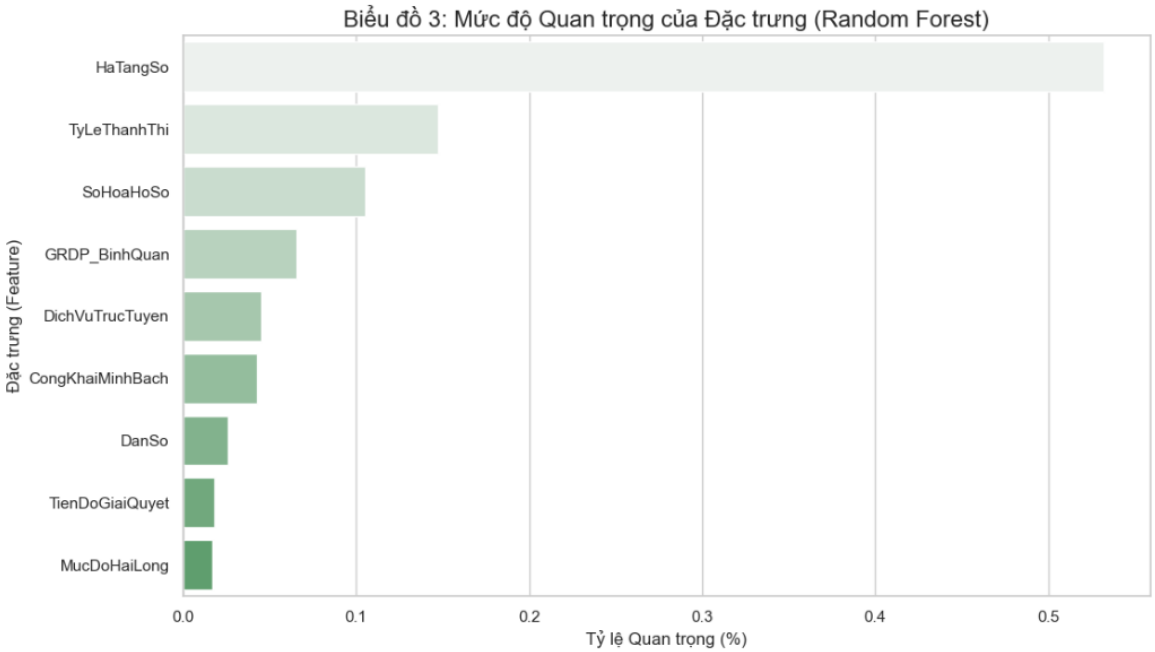


Hình 4.3.1 Biểu đồ so sánh hiệu quả của mô hình Linear Regression và Random Forest



Hình 4.3.2 Biểu đồ mức độ "Khớp" của 2 mô hình Linear Regression và Random Forest

Phát hiện Nổi bật: Phân tích Mức độ Quan trọng của Đặc trưng (Feature Importance)



Hình 4.3.3 Biểu đồ mức độ quan trọng của đặc trưng

Bảng Mức độ Quan trọng (từ cao đến thấp):

	Feature	Importance
0	HaTangSo	0.532132
3	TyLeThanhThi	0.147208
8	SoHoaHoSo	0.105482
2	GRDP_BinhQuan	0.065802
6	DichVuTrucTuyen	0.045057
4	CongKhaiMinhBach	0.043002
1	DanSo	0.026266
5	TienDoGiaiQuyet	0.018162
7	MucDoHaiLong	0.016887

Hình 4.3.4 Mức độ quan trọng của đặc trưng

Đây là kết quả giá trị nhất từ mô hình. Nó cho biết "bộ não" của mô hình "suy nghĩ" gì khi dự đoán DTI, và yếu tố nào là quan trọng nhất:

Kết luận Học thuật:

Mô hình đã chứng minh một cách định lượng rằng Hạ tầng số (HaTangSo) là yếu tố then chốt, quan trọng hơn tất cả các yếu tố khác cộng lại, trong việc thúc đẩy Chuyển đổi số cấp tỉnh.

CHƯƠNG 5: THIẾT KẾ ỨNG DỤNG (API & GIAO DIỆN)

Kiến trúc Backend sử dụng framework **FastAPI** và cơ sở dữ liệu **SQLite** để đảm bảo tính gọn nhẹ và dễ dàng triển khai.

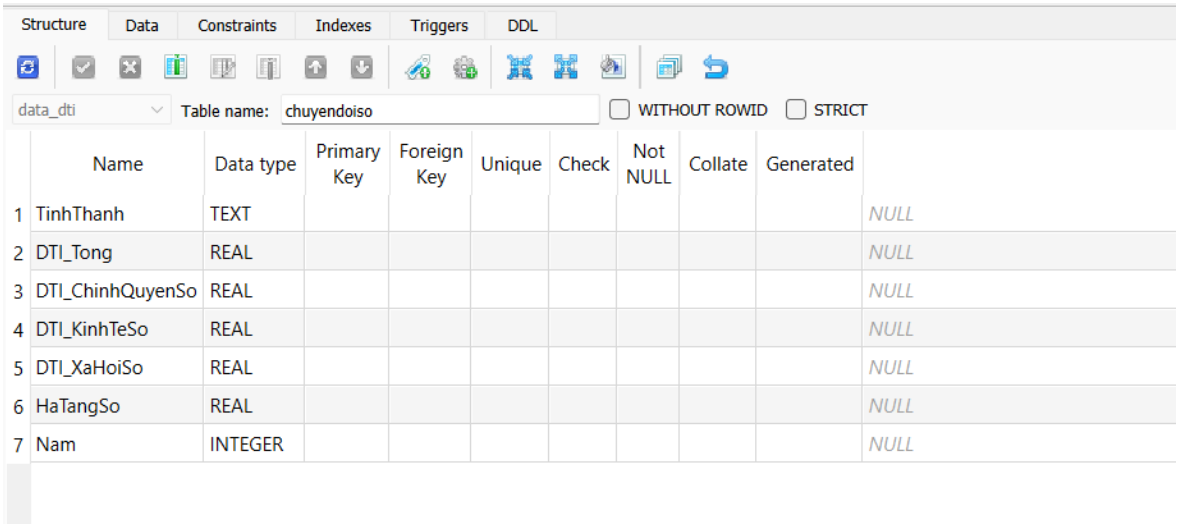
5.1. Cấu trúc cơ sở dữ liệu và API

5.1.1. Cấu trúc Cơ sở Dữ liệu (SQLite)

Hệ thống sử dụng SQLite làm nơi lưu trữ dữ liệu nguồn và dữ liệu đã hợp nhất.

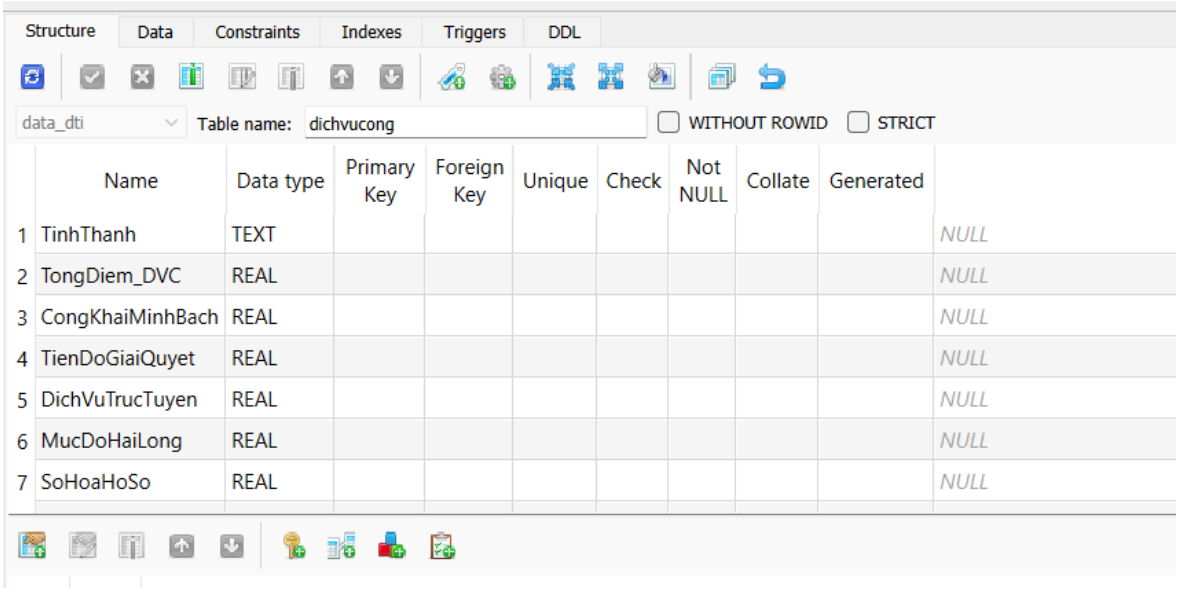
Bảng 5.1.1 Thông tin bảng trong database SQLite

Tên Bảng (Table)	**Mục đích**
chuyendoiso, ktxh, dichvucong	Lưu trữ dữ liệu thô/nguồn (để tái tạo master_df nếu cần).
tonghop	**Bộ dữ liệu cuối cùng (MASTER)**, chứa 16 đặc trưng đã hợp nhất.

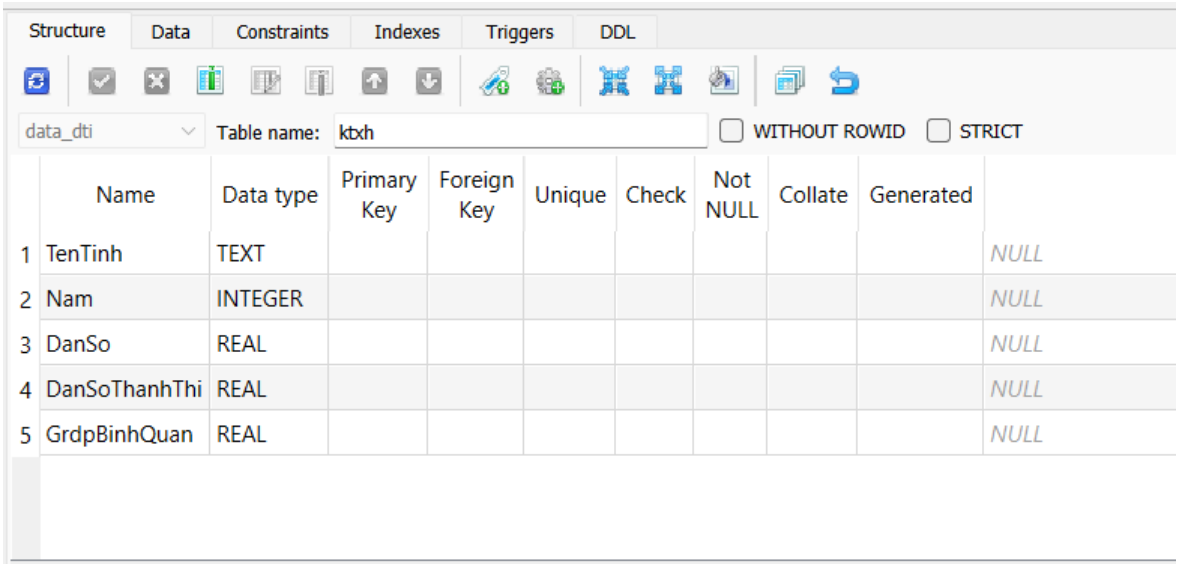


	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	TinhThanh	TEXT								NULL
2	DTI_Tong	REAL								NULL
3	DTI_ChinhQuyenSo	REAL								NULL
4	DTI_KinhTeSo	REAL								NULL
5	DTI_XaHoiSo	REAL								NULL
6	HaTangSo	REAL								NULL
7	Nam	INTEGER								NULL

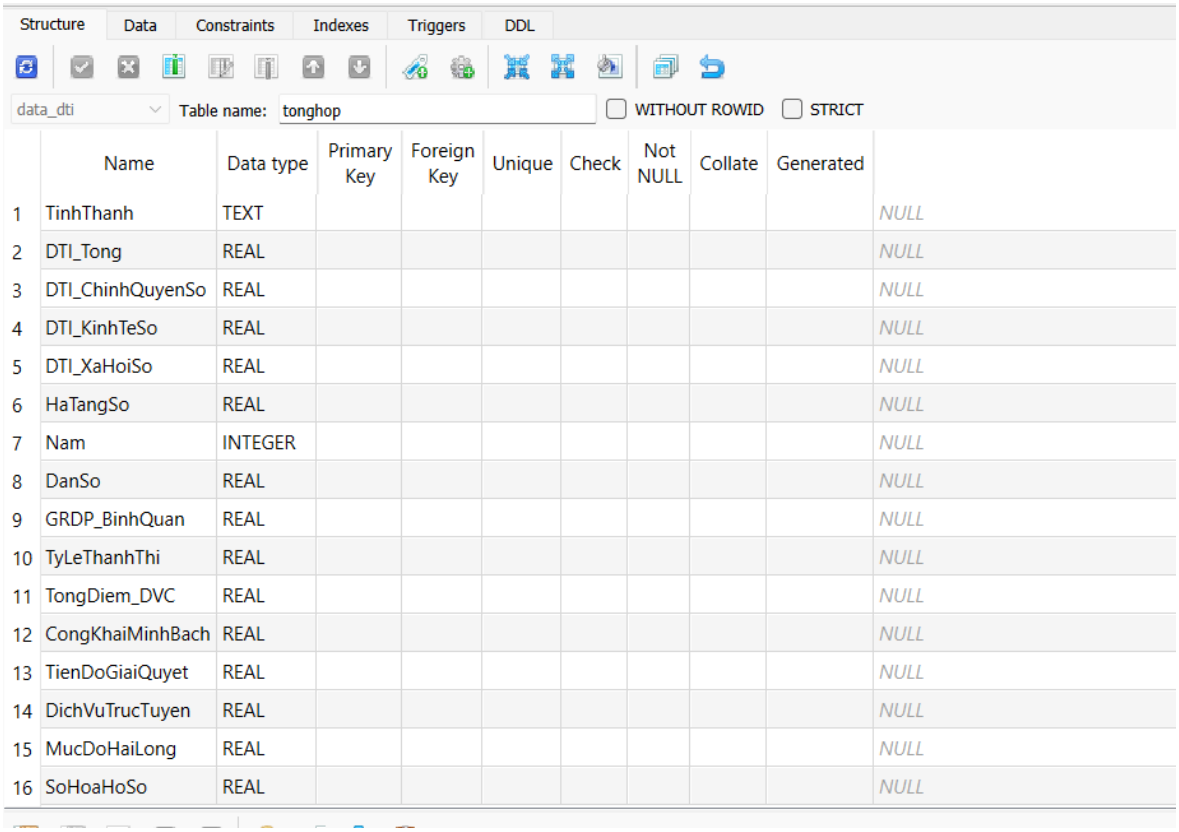
Hình 5.1.1 Chi tiết bảng chuyendoiso



Hình 5.1.2 Chi tiết bảng dichvucong



Hình 5.1.3 Chi tiết bảng ktxh



	Name	Data type	Primary Key	Foreign Key	Unique	Check	Not NULL	Collate	Generated	
1	TinhThanh	TEXT								NULL
2	DTI_Tong	REAL								NULL
3	DTI_ChinhQuyenSo	REAL								NULL
4	DTI_KinhTeSo	REAL								NULL
5	DTI_XaHoiSo	REAL								NULL
6	HaTangSo	REAL								NULL
7	Nam	INTEGER								NULL
8	DanSo	REAL								NULL
9	GRDP_BinhQuan	REAL								NULL
10	TyLeThanhThi	REAL								NULL
11	TongDiem_DVC	REAL								NULL
12	CongKhaiMinhBach	REAL								NULL
13	TienDoGiaiQuyet	REAL								NULL
14	DichVuTrucTuyen	REAL								NULL
15	MucDoHaiLong	REAL								NULL
16	SoHoaHoSo	REAL								NULL

Hình 5.1.4 Chi tiết bảng tonghop

5.1.2. Thiết kế API (FastAPI)

API được thiết kế tuân thủ nguyên tắc RESTful để truy vấn dữ liệu và dự đoán.

Bảng 5.1.2 Thông tin route API

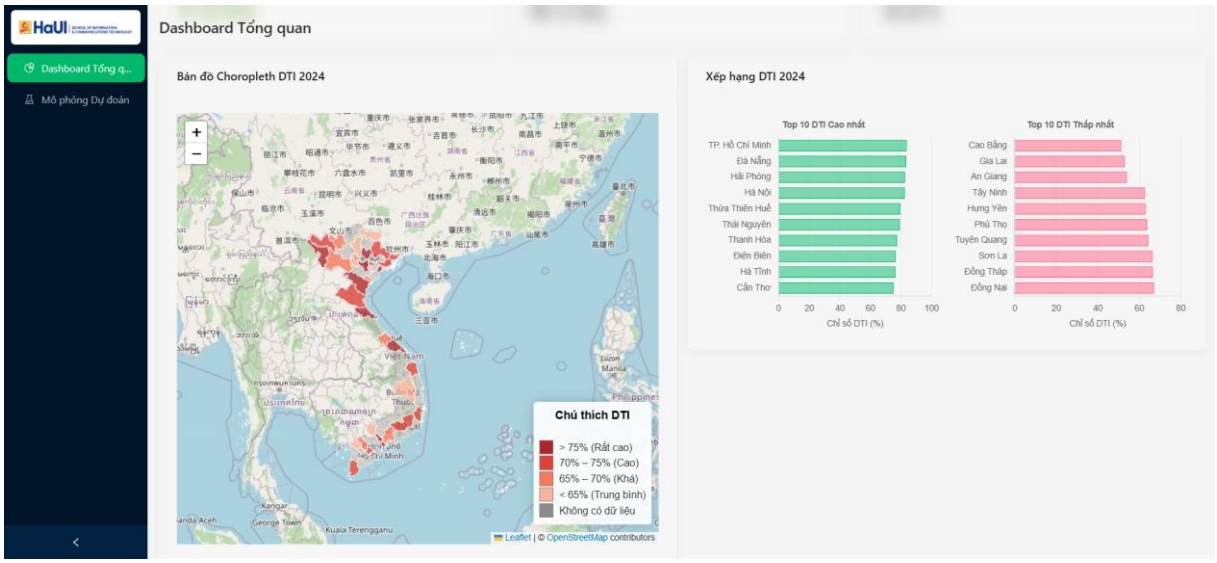
Endpoint	**Method**	**Chức năng**
/data-all	GET	Lấy toàn bộ 102 dòng dữ liệu hợp nhất.
/data/year/{year}	GET	Lọc dữ liệu theo Năm (cơ sở cho Giao diện Tổng quan).
/data/province/{province}	GET	Lấy dữ liệu chi tiết theo Tỉnh (cơ sở cho Giao diện Chi tiết).
/predict-dti	POST	Dự đoán chỉ số DTI (Nhận 9 đặc trưng thô, trả về DTI dự đoán).

5.2. Giới thiệu Chức năng Ứng dụng (Features)

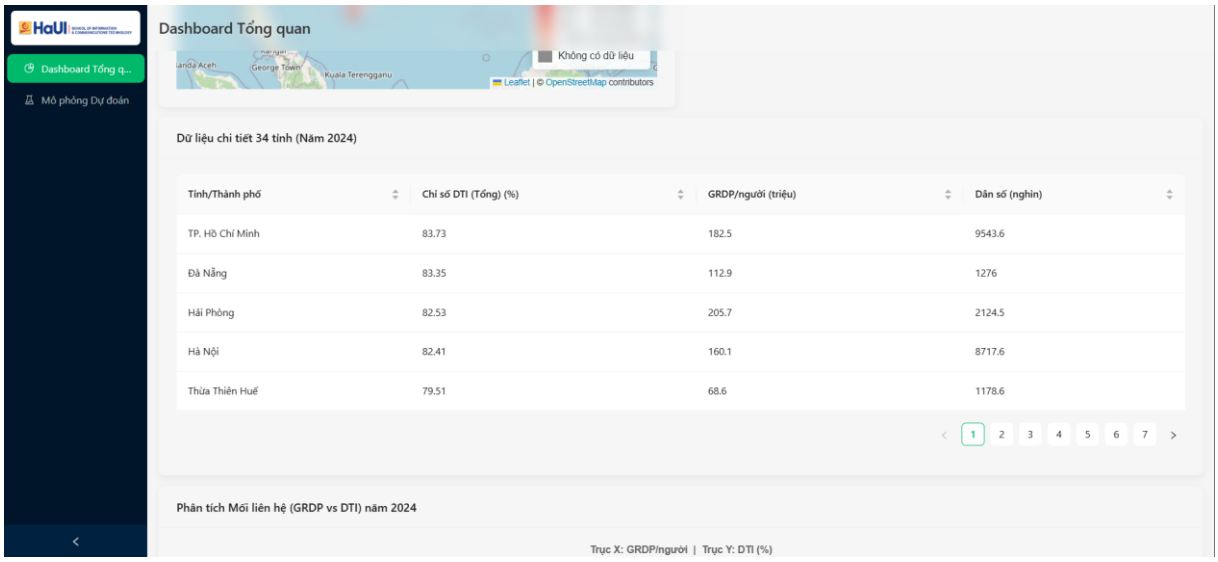
Giao diện Dashboard (React) được chia làm 3 khu vực chức năng chính:

5.2.1. Chức năng 1: Trang Tổng quan (So sánh 34 tỉnh)

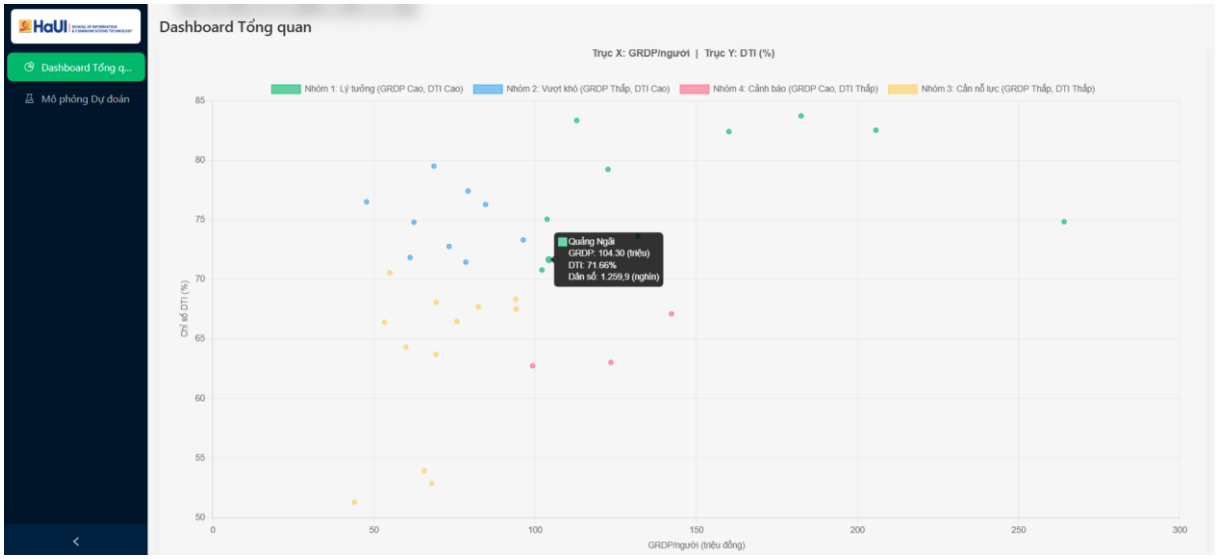
Hình 5.2.1 Trang tổng quan (1)



Hình 5.2.2 Trang tổng quan (2)



Hình 5.2.3 Trang tổng quan (3)



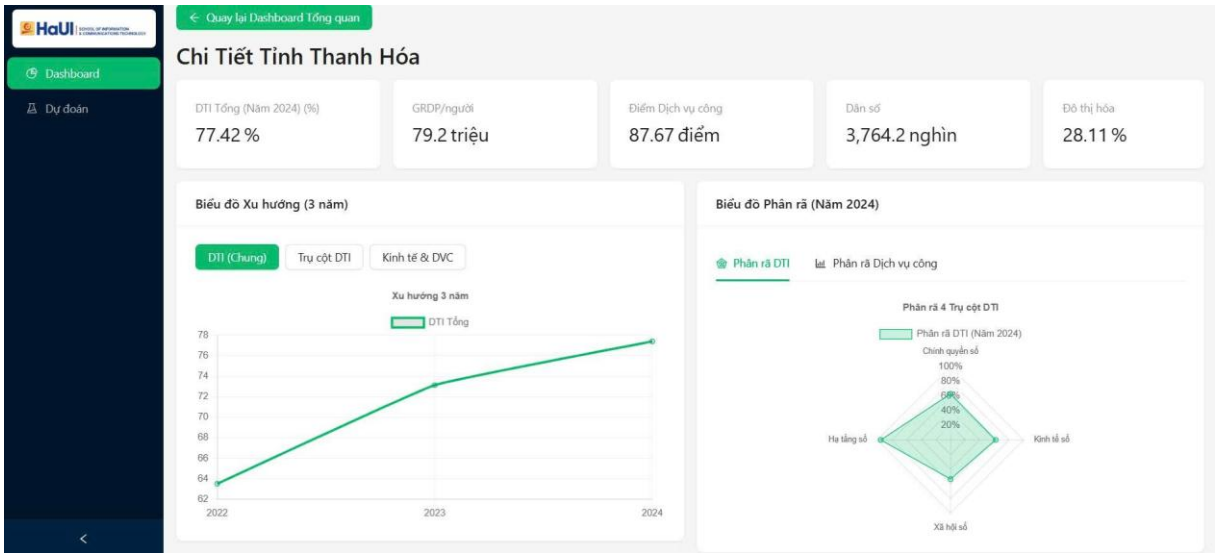
Hình 5.2.4 Trang tổng quan (4)

Mục tiêu: Cung cấp cái nhìn vĩ mô về 34 tỉnh/thành phố trọng điểm.

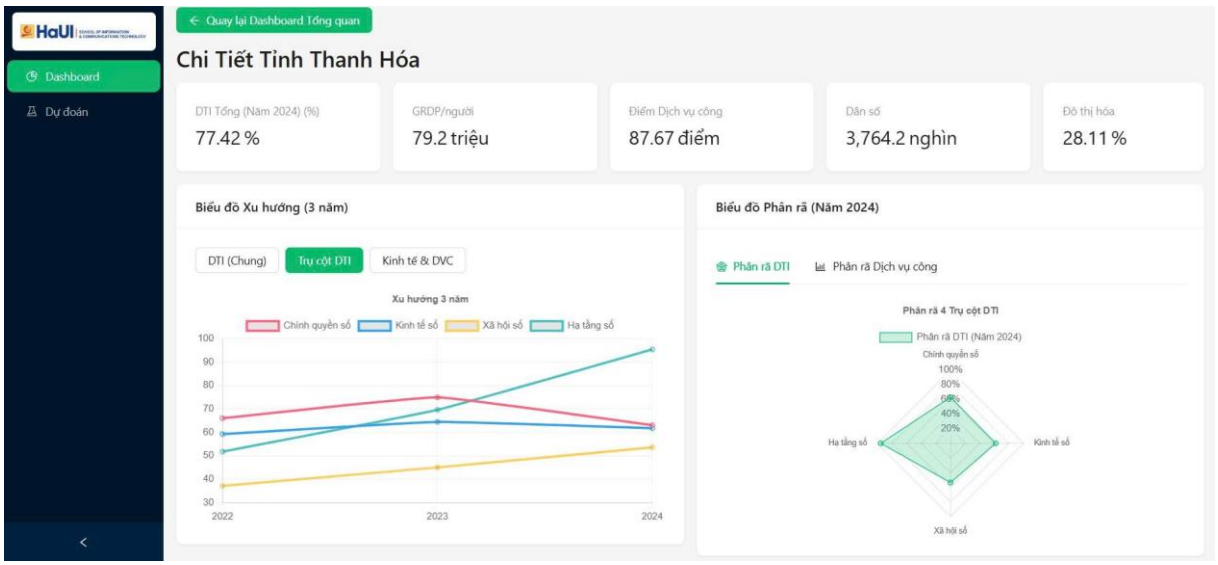
Tính năng:

- **Bản đồ Choropleth:** Trực quan hóa bản đồ Việt Nam, tô màu DTI_Tong cho 34 tỉnh.
- **Biểu đồ Tương quan:** Trực quan hóa mối liên hệ giữa GRDP_BinhQuan (Kinh tế) và DTI_Tong (CĐS).

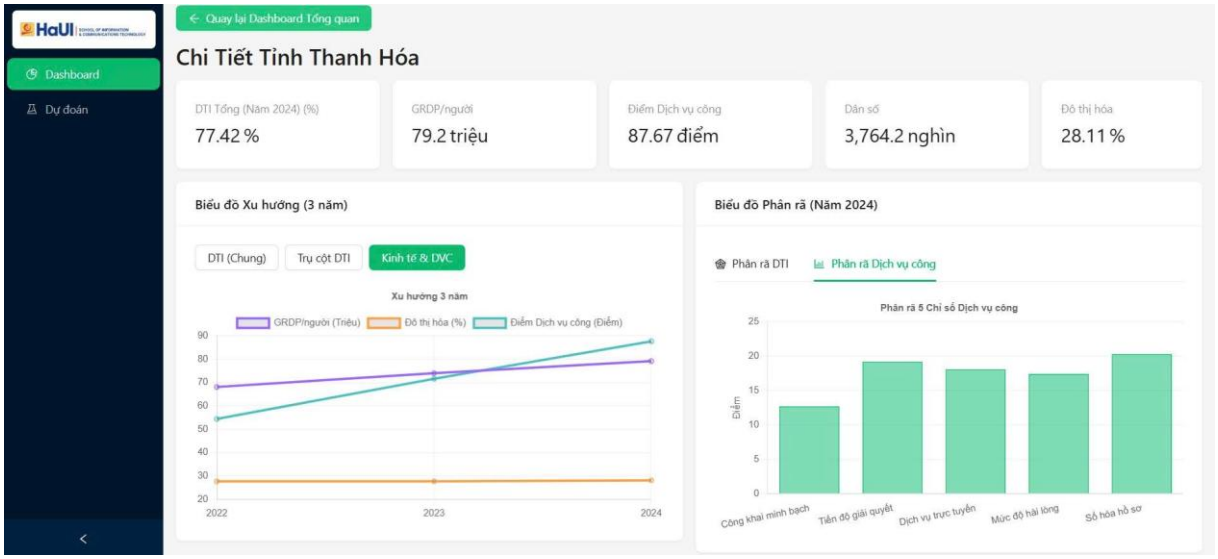
5.2.2. Chức năng 2: Trang Chi tiết Tỉnh



Hình 5.2.5 Trang chi tiết tỉnh (1)



Hình 5.2.6 Trang chi tiết tỉnh (2)



Hình 5.2.7 Trang chi tiết tỉnh (3)

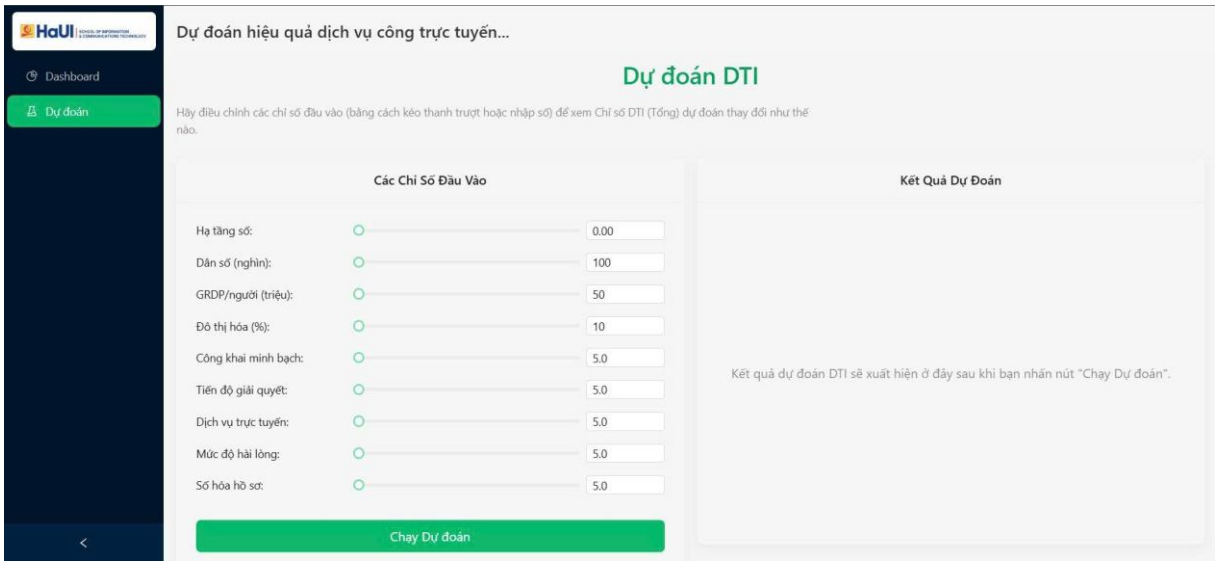
Mục tiêu: Cung cấp cái nhìn sâu về một tỉnh cụ thể.

Tính năng:

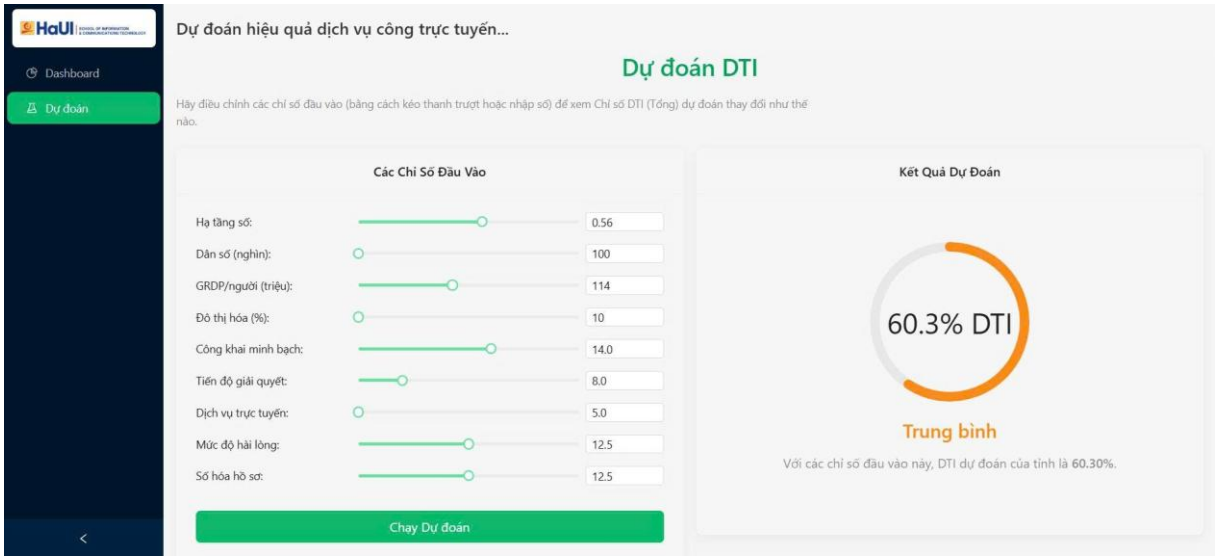
- **Thẻ KPI:** Hiển thị các chỉ số chính (DTI, GDP, DVC...) của tỉnh trong năm mới nhất**.**
- **Biểu đồ Xu hướng (Line Chart):** Theo dõi sự tăng trưởng của các chỉ số qua 3 năm.

- **Biểu đồ Phân rã (Tabs):** Phân tích 4 trụ cột DTI (Radar Chart) và 5 chỉ số DVC (Bar Chart).

5.2.3. Chức năng 3: Trang Mô phỏng Dự đoán ("What-if")



Hình 5.2.8 Trang dự đoán khi chưa dự đoán (1)



Hình 5.2.9 Trang dự đoán khi đã dự đoán (2)

Mục tiêu: Cho phép người dùng "thử nghiệm" kịch bản và xem tác động của các yếu tố lên Chuyển đổi số, dựa trên mô hình Random Forest đã huấn luyện.

Tính năng:

- **9 Thanh trượt (Sliders):** Người dùng kéo thả để thay đổi 9 giá trị đặc trưng đầu vào (Hạ tầng, Dân số, GDP...).

- **Đồng hồ đo (Gauge Chart):** Tự động cập nhật (real-time) bằng cách gọi API /predict-dti, hiển thị con số DTI dự đoán.

CHƯƠNG 6: TRIỂN KHAI VÀ GIẤY PHÉP

6.1. Giấy phép sử dụng và Lý do lựa chọn

Tại liệu nhóm tôi sử dụng tôi không tìm được giấy phép sử dụng nhưng Tất cả các nguồn được chọn đều là cơ quan nhà nước chính thức của Việt Nam (Bộ TT&TT, Tổng cục Thống kê, Văn phòng Chính phủ/Cổng DVC). Điều này đảm bảo dữ liệu có tính xác thực cao nhất và phù hợp để sử dụng trong một dự án nghiên cứu học thuật/cộng đồng.

Về luật pháp: Việc sử dụng các dữ liệu này cho mục đích phi thương mại, giáo dục và nghiên cứu được bảo vệ bởi **Luật Thống kê** và các quy định về Quản lý, kết nối và chia sẻ dữ liệu số (Nghị định 47/2020/NĐ-CP).

Vậy nên trong đề tài nghiên cứu này chúng tôi được phép sử dụng dữ liệu

Lý do lựa chọn: Vì các nguồn được gợi ý của đề thi mà nhóm cần thì đều không truy cập Đây là giấy phép tự do (permissive) đơn giản và phổ biến nhất, cho phép bất kỳ ai được toàn quyền sử dụng, sao chép, sửa đổi, và bán lại sản phẩm. Việc này khuyến khích tối đa sự chấp nhận, đóng góp và tái sử dụng của cộng đồng mã nguồn mở.

6.2. Kế hoạch Triển khai

Để chạy dự án trên máy cá nhân (localhost) cho mục đích phát triển và demo, chúng ta cần khởi chạy 2 tiến trình (process) độc lập: Backend API và Frontend Dashboard.

6.2.1. Yêu cầu Môi trường

- Python (ví dụ: 3.10+)
- Node.js (ví dụ: 18.x+) và npm.

6.2.2. Quy trình Chạy Backend (API Server)

- Mục tiêu: Khởi chạy máy chủ FastAPI trên cổng http://localhost:8000.
- Các bước:

1. Mở Terminal 1.
2. Di chuyển (cd) vào thư mục chứa file “API_Controller.py.”
3. Tạo và kích hoạt môi trường ảo Python:

   “python -m venv venv

   .\venv\Scripts\activate”
4. Cài đặt các thư viện (từ file requirements.txt):

   “pip install fastapi uvicorn pandas scikit-learn joblib pydantic fastapi-cors”
5. Cấu hình CORS (Bắt buộc): Sửa file API_Controller.py để cho phép Frontend (localhost:3000) gọi đến.

   “# Thêm vào đầu file API_Controller.py

   from fastapi.middleware.cors import CORSMiddleware

   # Thêm ngay sau khi tạo app = FastAPI()

   app.add_middleware(

   CORSMiddleware,

   allow_origins=["http://localhost:3000"], # Cho phép React gọi

   allow_credentials=True,

   allow_methods=["*"], # Cho phép tất cả methods (GET, POST)

   allow_headers=["*"],

)”
6. Khởi chạy Server:

   “uvicorn API_Controller:app --reload --port 8000”
7. Server Backend hiện đang chạy trên *“localhost:8000.”*

6.2.3. Quy trình Chạy Frontend (Dashboard)

- Mục tiêu: Khởi chạy máy chủ phát triển React trên cổng http://localhost:3000.
- Các bước:
    1. Mở Terminal 2 (Giữ nguyên Terminal 1).
    2. Di chuyển (cd) vào thư mục dự án React (frontend).
    3. Cài đặt thư viện (nếu là lần đầu):

       “npm install”
    4. Khởi chạy Server:

       “npm start”
    5. *Server Frontend hiện đang chạy trên localhost:3000.*
    6. Trình duyệt sẽ tự động mở trang http://localhost:3000. Dashboard sẽ tải và gọi API đến localhost:8000 để lấy dữ liệu.

CHƯƠNG 7: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

7.1. Kết luận

Dự án đã xây dựng thành công một pipeline dữ liệu hoàn chỉnh, từ việc thu thập 3 nguồn dữ liệu thô (DTI, GSO, DVC) đến việc tạo ra một bộ dữ liệu sạch (master_df - 102 dòng) của 34 tỉnh/thành phố trong 3 năm.

Bằng cách áp dụng mô hình Random Forest (đã tối ưu), dự án đã định lượng được các yếu tố ảnh hưởng đến Chuyển đổi số. Phát hiện quan trọng nhất là **Hạ tầng số (HaTangSo)** là yếu tố dự đoán quan trọng áp đảo, chiếm hơn 50% tầm quan trọng trong mô hình.

Cuối cùng, dự án đã thiết kế một kiến trúc ứng dụng web (Flask + React + Docker) hoàn chỉnh, cung cấp 3 chức năng tương tác cốt lõi: **Tổng quan (Bản đồ)**, **Chi tiết (Xu hướng)**, và **Dự đoán (Mô phỏng "What-if")**.

7.2. Hướng phát triển (Mã nguồn mở)

Mở rộng Dữ liệu (Rows):

- Tích cực "săn" thêm dữ liệu lịch sử (các năm 2020, 2021).
- Theo dõi Cổng DVC (dichvucong.gov.vn) để cập nhật khi Bảng 3 có thêm dữ liệu cho 29 tỉnh còn lại.

Mở rộng Đặc trưng (Features):

- Bổ sung các đặc trưng mới để tăng độ chính xác của mô hình, ví dụ:
- Tỷ lệ phủ sóng 4G/5G (từ Bộ TT&TT).
- Chỉ số Hiệu quả Quản trị và Hành chính công (PAPI).
- Số lượng doanh nghiệp CNTT (từ Niên giám Thống kê GSO).

Triển khai Công khai (Public Deployment):

- Đẩy mã nguồn hoàn chỉnh của Backend và Frontend lên GitHub với Giấy phép MIT.
- Triển khai ứng dụng lên các nền tảng miễn phí (như Render.com, Vercel, hoặc Hugging Face Spaces) để cộng đồng có thể truy cập và sử dụng.

Kế hoạch Triển khai (Docker)

- Để đảm bảo tính nhất quán và dễ dàng triển khai, ứng dụng Backend API được đóng gói bằng Docker.
- Dockerfile: Được tạo để định nghĩa môi trường chạy cho ứng dụng (sử dụng image python:3.10-slim).
- Server Production: Sử dụng Gunicorn làm máy chủ WSGI production, đảm bảo API có thể xử lý nhiều yêu cầu đồng thời một cách ổn định.
- docker-compose.yml (Tương lai): Sẽ được sử dụng để khởi chạy 2 containers (backend và frontend) cùng lúc, tạo thành một hệ thống tích hợp hoàn chỉnh.

TÀI LIỆU THAM KHẢO

[1] Giáo trình Trí tuệ nhân tạo, Trường Đại học Công Nghiệp Hà Nội.

[2] Đỗ Ngọc Sơn, Phan Văn Viên, Nguyễn Phương Nga (2015), *Giáo trình Hệ quản trị cơ sở dữ liệu*, NXB Khoa học và Kỹ thuật.

[3] https://dti.gov.vn/

[4] https://dichvucong.gov.vn/p/home/dvc-index-tinhthanhpho-dvctructuyen.html

[5] T. A. Dao and D. C. Nguyen, "Digital Transformation in Vietnam: Policies, Results and Recommendations," Journal of Southeast Asian Economies (JSEAE), vol. 40, no. 1, pp. 74-99, 2023. DOI: 10.1355/ae40-1f.

[6] B. Alhayani, A. S. Corallo, et al., "Automating E-Government Services With Artificial Intelligence," IEEE Access, vol. 10, pp. 25056-25080, 2022. DOI: 10.1109/ACCESS.2022.3155169.

[7] N. T. T. Huyen, "Digital Transformation in Public Administration in Vietnam: Current Status, Challenges, and Policy Implications," VNU Journal of Science: Policy and Management Studies, vol. 41, no. 1, 2025.