

BÁO CÁO CUỐI KỲ
CÔNG NGHỆ DỮ LIỆU LỚN
**HUẤN LUYỆN MÔ HÌNH DỮ LIỆU LỚN
ĐỂ PHÂN LOẠI BẤT ĐỘNG SẢN CHO
THUÊ**

GVHD: TS. Đỗ Trọng Hợp

Lớp: IE212.P11.VB2

Nhóm sinh viên thực hiện:

Nguyễn Xuân Thanh (Nhóm trưởng) - 21540020

Phạm Ngọc Hoàng - 22540015

Phan Trung Phong - 22540007

Tp. Hồ Chí Minh, 2/2025

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

....., ngày tháng năm 2025

Người nhận xét

(Ký tên và ghi rõ họ tên)

PHÂN CÔNG CÔNG VIỆC

Phân công công việc trong nhóm được thực hiện như sau:

Nguyễn Xuân Thanh (MSSV: 22540020), với vai trò nhóm trưởng, chịu trách nhiệm lập đề tài, lên ý tưởng cho báo cáo, xây dựng sườn báo cáo, viết các chương 2, 3, 4, 5, 6, 7 cho báo cáo words và powerpoint. Ngoài ra, anh còn viết toàn bộ source code cào data với 8,554 dòng từ website Mogi trên VSC, thực hiện tiền xử lý dữ liệu và dán nhãn trên Colab, huấn luyện mô hình trên Colab, cũng như streaming dữ liệu bằng Kafka và Spark trên Pycharm, và đã hoàn thành đầy đủ 100%.

Phạm Ngọc Hoàng (MSSV: 22540015) đóng góp ý tưởng cho báo cáo và đảm nhận việc viết Chương 1 về Đặt vấn đề, với kết quả hoàn thành đầy đủ 100%.

Phan Trung Phong (MSSV: 22540007) cũng đóng góp ý tưởng cho báo cáo và chịu trách nhiệm viết Chương 6 về Kết luận, đạt mức hoàn thành 100%.

LỜI CẢM ƠN

Để hoàn thành tốt đề tài đồ án môn học, chúng em đã nhận được rất nhiều sự hỗ trợ, đóng góp tích cực đến từ quý Thầy Đỗ Trọng Hợp và các bạn cùng lớp. Theo đó, chúng em đã tiếp thu và vận dụng tối đa các kiến thức được học, các góp ý, phản hồi một cách đầy đủ nhất để hoàn thành đề tài này theo đúng mong đợi của mình.

Trước hết, chúng em xin gửi lời cảm ơn đến quý Thầy Cô trường Đại học Công nghệ Thông tin, đặc biệt là quý Thầy Cô khoa Khoa học và Kỹ thuật Thông tin. Những kiến thức quý báu mà quý Thầy cô đã truyền đạt cho chúng em trong quá trình học tập và rèn luyện tại trường chính là những giá trị giúp chúng em không ngừng hoàn thiện khả năng của bản thân, đồng thời giúp chúng em trang bị những kỹ năng cần thiết để hoàn thành đề tài đồ án một cách tốt nhất.

Chúng em cũng xin gửi lời cảm ơn đặc biệt chân thành tới TS. Đỗ Trọng Hợp - người Thầy đã đồng hành cùng chúng em trong suốt thời gian thực hiện đồ án môn học. Sự chỉ dẫn nhiệt tình và những góp ý của Thầy đã giúp chúng em xác định hướng đi đúng đắn, nắm vững kiến thức chuyên môn và có thể vượt qua các thách thức trong quá trình triển khai đề tài. Sự tận tâm và lòng nhiệt huyết của Thầy đã truyền cảm hứng cho chúng em và giúp chúng em phát triển không chỉ về mặt kiến thức mà còn về kỹ năng làm việc nhóm và giải quyết vấn đề.

Một lần nữa, nhóm em xin gửi lời cảm ơn sâu sắc nhất đến nhà trường, quý Thầy Cô và các bạn đã tạo điều kiện để nhóm có được một trải nghiệm học tập bổ ích và các kiến thức đầy đủ đáp ứng được các yêu cầu của chuyên ngành học.

LỜI MỞ ĐẦU

Trong bối cảnh kinh tế hiện nay, ngành bất động sản luôn đóng vai trò then chốt trong việc phát triển kinh tế – xã hội của một quốc gia. Sự biến động của thị trường, sự thay đổi nhu cầu và khả năng chi trả của khách hàng tạo nên một môi trường cạnh tranh khốc liệt. Chính vì vậy, việc định hướng và phân loại khách hàng theo mức thu nhập đóng vai trò quan trọng trong chiến lược kinh doanh, giúp các nhà môi giới và doanh nghiệp bất động sản có thể đưa ra các chính sách bán hàng, cho thuê cũng như định giá phù hợp.

Trên nền tảng công nghệ hiện đại, dữ liệu được thu thập từ các website đăng tin bất động sản như MOGI cung cấp một nguồn thông tin phong phú về các chỉ số của bất động sản: giá, diện tích, số phòng ngủ, vị trí, cũng như các tiện ích xung quanh như bệnh viện, trường học, siêu thị... Những thông tin này không chỉ giúp khách hàng có cái nhìn tổng quan về thị trường mà còn mở ra cơ hội cho các doanh nghiệp xây dựng các hệ thống dự đoán, phân loại khách hàng dựa trên mức thu nhập.

Bài báo cáo này trình bày chi tiết quá trình xây dựng và triển khai hệ thống dự đoán nhân thu nhập dựa trên dữ liệu bất động sản được thu thập từ website MOGI. Hệ thống được xây dựng trên nền tảng Apache Spark Structured Streaming kết hợp với Kafka để xử lý dữ liệu thời gian thực, đồng thời tích hợp mô hình học máy (Random Forest Pipeline) đã được huấn luyện trước nhằm phân loại khách hàng thành các nhóm như: “người thu nhập thấp”, “người trẻ độc thân thu nhập trung bình”, “người trẻ độc thân thu nhập khá”, “người độc thân thu nhập cao”, “gia đình thu nhập thấp”, “gia đình thu nhập cao”.

Mục tiêu của hệ thống là hỗ trợ các nhà phân tích và doanh nghiệp trong việc xác định đối tượng khách hàng mục tiêu, từ đó tối ưu hóa các chiến lược tiếp thị và chăm sóc khách hàng. Qua quá trình xử lý, phân tích và dự đoán, hệ thống không chỉ giúp đưa ra các nhận định chính xác mà còn thể hiện được hiệu năng xử lý dữ liệu thời gian thực, góp phần cải thiện quá trình ra quyết định trong kinh doanh.

Bài báo cáo sẽ đi sâu vào từng khía cạnh của dự án, từ việc thu thập dữ liệu, tiền xử lý, xây dựng mô hình, cho đến đánh giá hiệu quả và triển khai hệ thống dự đoán nhân thu nhập. Qua đó, báo cáo mong muốn cung cấp một cái nhìn tổng quan cũng như chi tiết về cách thức vận hành và ứng dụng của hệ thống trong thực tiễn, góp phần nâng cao hiệu quả quản lý và kinh doanh trong lĩnh vực bất động sản.

Với nền tảng của các công nghệ xử lý dữ liệu lớn và học máy tiên tiến, hệ thống dự đoán nhân thu nhập không chỉ là một công cụ hỗ trợ đắc lực cho các chuyên gia bất động sản mà còn mở ra hướng đi mới trong việc ứng dụng trí tuệ nhân tạo vào lĩnh vực phân tích và định giá bất động sản.

ĐẶT VẤN ĐỀ

Bối cảnh

Trong bối cảnh kinh tế hiện nay, ngành bất động sản đóng vai trò then chốt trong phát triển kinh tế – xã hội. Thị trường liên tục biến động vì nhu cầu và khả năng chi trả của khách hàng luôn thay đổi. Việc phân loại khách hàng theo mức thu nhập hoặc đặc điểm kinh tế – xã hội là vô cùng quan trọng, giúp các doanh nghiệp bất động sản đưa ra chiến lược định giá, marketing và chăm sóc phù hợp.

Lý do chọn đề tài

- Thứ nhất, các website đăng tin bất động sản (như Mogi) sở hữu lượng dữ liệu lớn, giàu thông tin về giá, diện tích, vị trí, tiện ích xung quanh,...
- Thứ hai, nhu cầu xử lý dữ liệu này trong thời gian thực (real-time streaming) ngày càng tăng, đòi hỏi tích hợp công nghệ Apache Kafka và Apache Spark để dễ dàng mở rộng và đáp ứng tốc độ.
- Thứ ba, việc dự đoán xem bất động sản đang được rao (với các đặc trưng nhất định) sẽ phù hợp với nhóm đối tượng thuê nhà nào (gia đình, người độc thân, thu nhập thấp/cao,...) mang lại lợi ích thiết thực cho cả người đăng tin lẫn người tìm thuê.

Mục tiêu

- Xây dựng một hệ thống thu thập dữ liệu bất động sản từ Mogi, tiến hành tiền xử lý, gán nhãn dựa trên thu nhập/đặc điểm người thuê.
- Huấn luyện mô hình học máy (Machine Learning) để dự đoán nhóm đối tượng thuê nhà tiềm năng.
- Tích hợp Kafka và Spark Streaming để xử lý và dự đoán dữ liệu thời gian thực.

BÀI TOÁN ĐẶT RA

Mô tả tổng quan

Bài toán: Dự đoán “nhân” (hay phân khúc) bất động sản cho người thuê nhà dựa vào đặc trưng của bất động sản. Ví dụ một tin đăng với giá thuê, diện tích, vị trí, tiện ích xung quanh,... sẽ tương ứng với nhóm đối tượng phù hợp:

- Người độc thân thu nhập thấp/trung bình/khá/cao
- Gia đình thu nhập thấp/cao



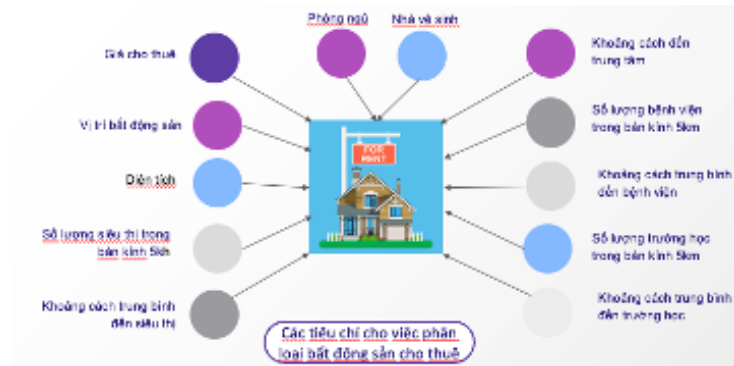
Hình 1: Dự đoán “nhân” (hay phân khúc) bất động sản dựa vào đặc trưng của bất động sản

Ý nghĩa

- Giúp người môi giới/doanh nghiệp bất động sản xác định khách hàng mục tiêu nhanh hơn.
- Tối ưu quá trình marketing, đề xuất gói sản phẩm phù hợp.
- Cải thiện trải nghiệm người thuê nhà, giúp họ nhanh chóng chọn được bất động sản ưng ý.

Yêu cầu xử lý

- Tiền xử lý dữ liệu: Chuyển đổi đơn vị (giá, diện tích), xóa giá trị khuyết (NaN), phát hiện ngoại lai (outlier).
- Chia dữ liệu train/test: Tỷ lệ 7:3, đảm bảo tính đại diện.
- Xây dựng mô hình: Gợi ý sử dụng Random Forest hoặc mô hình khác (XGBoost, v.v.) để phân loại.
- Triển khai streaming: Tích hợp dữ liệu mới từ Kafka, dùng Spark Streaming để áp mô hình dự đoán liên tục.



Hình 2: Các tiêu chí cho việc phân loại bất động sản cho thuê



Hình 3: Quy trình xử lý phân loại bất động sản

CÔNG NGHỆ

Apache Kafka

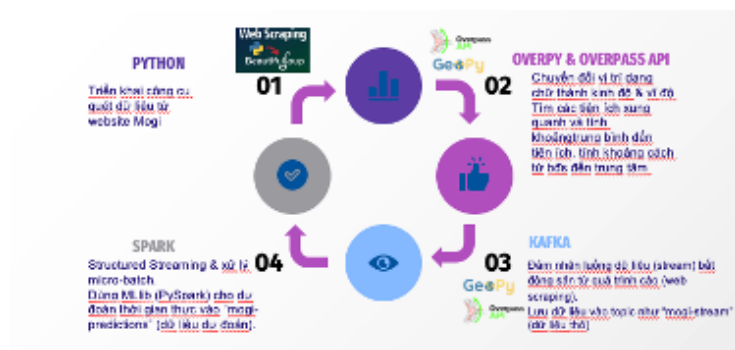
- Đảm nhận luồng dữ liệu (stream) bất động sản từ quá trình cào (web scraping).
- Lưu dữ liệu vào các topic như “mogi-stream” (dữ liệu thô) và “mogi-predictions” (dữ liệu dự đoán).

Apache Spark

- Spark Structured Streaming: Đọc dữ liệu từ Kafka, xử lý theo micro-batch hoặc continuous mode.
- Spark MLlib (hoặc PySpark ML) dùng để load mô hình Random Forest Pipeline đã huấn luyện và chạy dự đoán trong thời gian thực.

Cơ sở dữ liệu và dịch vụ khác

- GeoPy và Overpass API hỗ trợ tính toán khoảng cách đến các tiện ích xung quanh (bệnh viện, trường học,...).

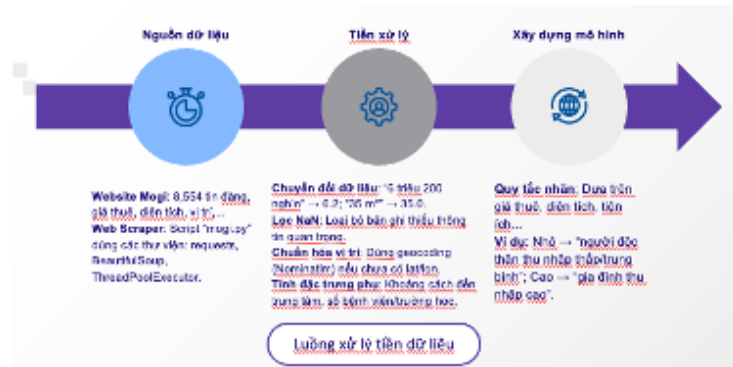


Hình 4: Các công nghệ sử dụng

DATASET

Nguồn dữ liệu

- Website Mogi: Thu thập khoảng 8,554 tin đăng, chứa các trường quan trọng như giá thuê (VND), diện tích (m²), vị trí (lat, lon), số phòng ngủ/tắm, ngày đăng, v.v.
- Dữ liệu được cào (crawl) thông qua script “web_scraper_kafka.py” dùng requests và BeautifulSoup.



Hình 5: Luồng xử lý tiền dữ liệu

Tiền xử lý

- Chuyển đổi kiểu dữ liệu: từ chuỗi “6 triệu 200 nghìn” thành số thực 6.2 (triệu VND), từ “35 m²” thành 35.0 (m²).
- Lọc bỏ NaN: thiếu trường quan trọng thì bỏ bản ghi hoặc áp dụng kỹ thuật bổ khuyết (imputation).
- Chuẩn hóa vị trí: vì chưa có kinh độ/vĩ độ, có thể dùng hàm geocoding (Nominatim) để lấy.
- Tính các đặc trưng phụ: khoảng cách đến trung tâm, số bệnh viện/trường học trong bán kính 5 km.

Gán nhãn (Label)


- Dựa trên giá thuê, diện tích, tiện ích,... để tự định nghĩa quy tắc gán nhãn phân khúc người thuê.
- Ví dụ: Nếu giá thuê và diện tích nhỏ, có thể là “người độc thân thu nhập thấp/trung bình”. Nếu giá thuê cao, diện tích rộng, nhiều tiện ích, có thể là “gia đình thu nhập cao”.

Hình 8: Gán nhãn (Labeling) cho bất động sản

HUẤN LUYỆN

Chia dữ liệu Train/Test (7:3)

- Train set (70%): Huấn luyện mô hình.
- Test set (30%): Đánh giá mô hình.
- Dữ liệu: File CSV (khoảng 8554 dòng), chứa thông tin: Phòng ngủ, WC, vị trí, giá (triệu VND), diện tích (m²), tọa độ, khoảng cách đến trung tâm, số bệnh viện/trường học, v.v. Label: phân loại đối tượng (6 nhóm).



```
[5]: train_df, test_df = df_ugars.randomSplit([0.7, 0.3], seed=0)

print("Train count:", train_df.count())
print("Test count:", test_df.count())
```

Train count: 6026
Test count: 2478

Hình 9: Chia dữ liệu Train/Test (7:3)

Xây dựng mô hình

- Random Forest Pipeline (trong PySpark ML):
 - StringIndexer (nếu có cột dạng chuỗi cần chuyển sang số).
 - VectorAssembler (gộp các đặc trưng dạng số vào cột features).
 - RandomForestClassifier (hoặc RandomForestRegressor).
- Tham số: Số cây (numTrees), độ sâu (maxDepth), v.v.
- Thư viện:
 - PySpark ML (Xử lý, mô hình, pipeline)
 - Scikit-learn (ROC, AUC), seaborn, matplotlib (vẽ biểu đồ)
 - Pandas (chuyển đổi từ Spark DataFrame)

Luồng xử lý:

- Đọc dữ liệu: `spark.read.csv(...)` → Spark DataFrame
- Pipeline: `Pipeline(stages=[indexer, assembler, rf])`
- Huấn luyện: `model = pipeline.fit(train_df)`
- Dự đoán: `predictions = model.transform(test_df)`

Đánh giá:

- Accuracy $\approx 98.6\%$ (2444/2478 mẫu đúng).

```

from pyspark.ml.feature import StringIndexer, VectorAssembler

# Cột numeric (features) - KHÔNG có final_score
feature_cols = [
    "price (million VND)",
    "restroom",
    "bedroom",
    "square (m2)",
    "distance_to_center",
    "hospital_count",
    "school_count",
    "hospital_avg_distance",
    "school_avg_distance",
    "super_market_count",
    "super_market_avg_distance"
]

# Bước 1: index label
indexer = StringIndexer(inputCol="label", outputCol="labelIndex")

# Bước 2: assemble features
assembler = VectorAssembler(inputCols=feature_cols, outputCol="features")

```

Hình 10: PySpark ML

```

| | import pandas as pd
| | import matplotlib.pyplot as plt
| | import seaborn as sns
| |
| | # Nhóm dữ liệu dự đoán theo (labelIndex, prediction), lấy số lượng
| | conf_spark = predictions.groupby(["labelIndex", "prediction"]).count()
| |
| | # Chuyển DataFrame Spark thành Pandas
| | conf_pdf = conf_spark.toPandas()
| |
| | # Pivot: biến cột "prediction" thành các cột, "labelIndex" thành các dòng
| | conf_matrix = conf_pdf.pivot(index="labelIndex", columns="prediction", values="count").fillna(0)
| |
| | # Vẽ heatmap
| | plt.figure(figsize=(8, 6))
| | sns.heatmap(conf_matrix, annot=True, cmap="Blues", fmt=".0f")
| | plt.xlabel("Predicted label")
| | plt.ylabel("True Label")
| | plt.title("Confusion Matrix")
| | plt.show()

```

Hình 11: Scikit-learn (ROC, AUC) - seaborn, matplotlib (vẽ biểu đồ) - pandas (chuyển đổi từ Spark DataFrame)

- F1 Score (tổng quát, đa lớp) cũng rất cao ($\approx 0.98+$).
- ROC, AUC cho từng lớp (one-vs-rest) $\sim 0.98-0.99$ (tùy lớp).

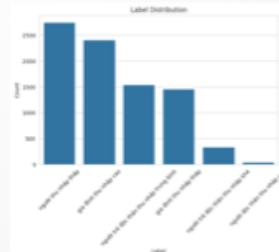
Triển khai:

- Lưu mô hình: `model.write().save(...)`
- Dùng `model.transform(new_df)` cho bản ghi mới (Spark DataFrame 1 dòng).
- Lấy nhãn dự đoán bằng `indexer_model.labels[prediction]`.

Kết luận:

- Mô hình Random Forest + Pipeline PySpark hoạt động tốt, độ chính xác

5.2 Xây dựng mô hình
- Random Forest Pipeline (trong PySpark ML):
Label Distribution
• Biểu đồ cột thể hiện số lượng mẫu thuộc mỗi nhóm (như "người thu nhập thấp", "gia đình thu nhập cao", v.v.).

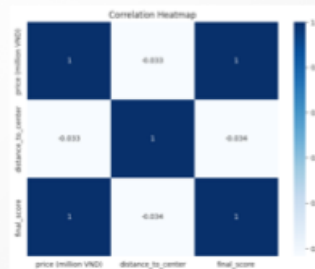


Label Distribution

Hình 15: Label Distribution

- Các giá trị khá thấp (gần 0), tức là không có mối tương quan mạnh giữa các biến này.

5.2 Xây dựng mô hình
- Random Forest Pipeline (trong PySpark ML):
Correlation Heatmap
• Ma trận tương quan giữa các biến (price, distance_to_center, final_score).
• Các giá trị khá thấp (gần 0), tức là không có mối tương quan mạnh giữa các biến này.



Correlation Heatmap

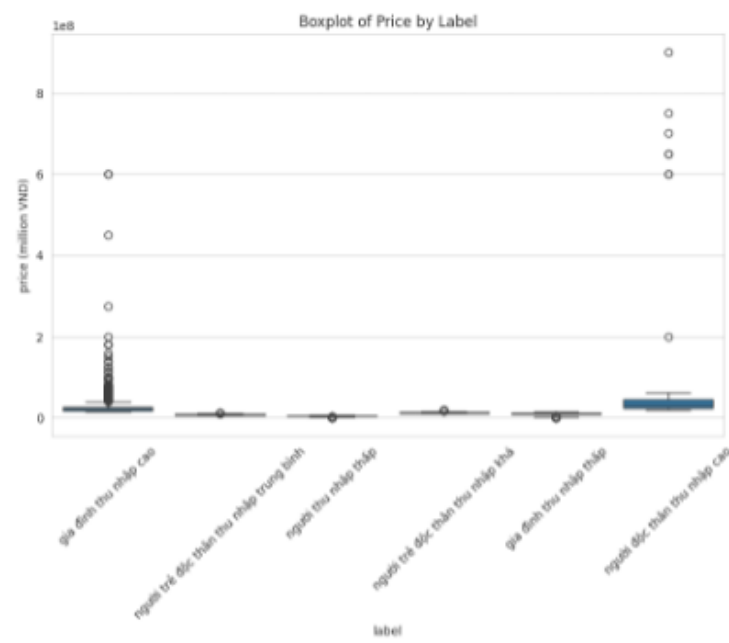
Hình 16: Correlation Heatmap

Boxplot of Price by Label:

- Hộp thể hiện phân bố giá (price) theo từng nhóm nhãn.
- Cho thấy sự chênh lệch lớn về giá giữa các nhóm, với nhiều giá trị ngoại lệ (outliers).

Distribution of Price:

- Biểu đồ phân bố (histogram+kde) của giá bất động sản (price).
- Dữ liệu lệch phải mạnh (right-skewed), phần lớn giá ở mức thấp, một số giá trị cực cao.



Hình 17: Boxplot of Price by Label

Confusion Matrix (số index):

- Ma trận nhầm lẫn hiển thị tần suất nhãn thực (hàng) và nhãn dự đoán (cột) bằng chỉ số (0,1,2,...).
- Các giá trị chéo cho thấy mô hình phân loại đúng.

Confusion Matrix (tên nhãn):

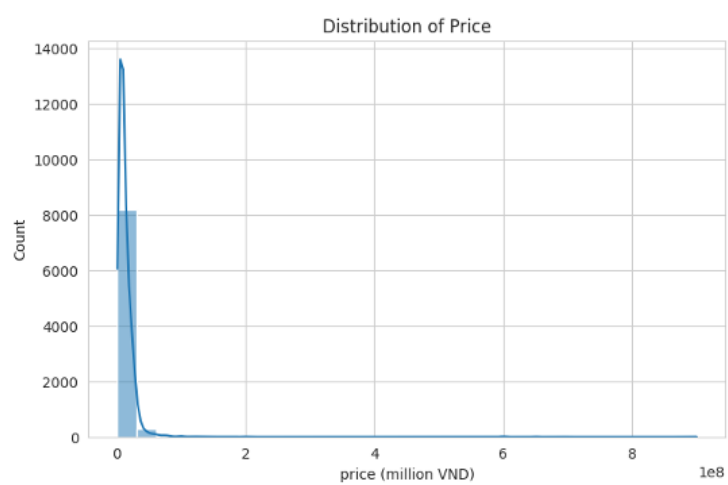
- Tương tự ma trận nhầm lẫn ở trên, nhưng hiển thị tên nhãn gốc (ví dụ “người thu nhập thấp”, “gia đình thu nhập cao”,...).
- Giúp dễ quan sát lớp nào bị dự đoán nhầm thành lớp nào.

ROC Curve - Multi-class (One-vs-Rest):

- Vẽ đường ROC cho từng lớp (coi lớp đó là “dương” và các lớp còn lại là “âm”).
- Đường cong sát trục trên cho thấy mô hình phân biệt rất tốt cho từng lớp (AUC=1).

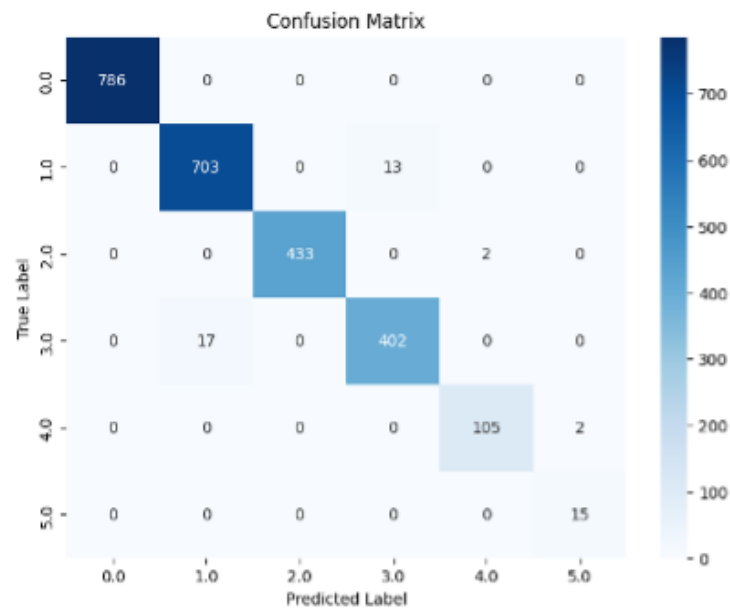
Lưu mô hình

- Sau khi huấn luyện xong, mô hình được lưu ra thư mục (“rf_model_pipeline”).

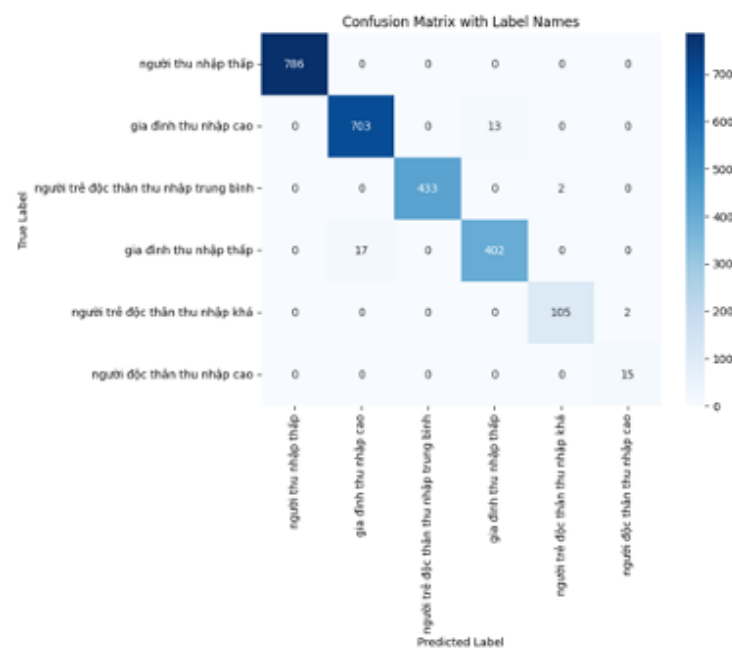


Hình 18: Distribution of Price

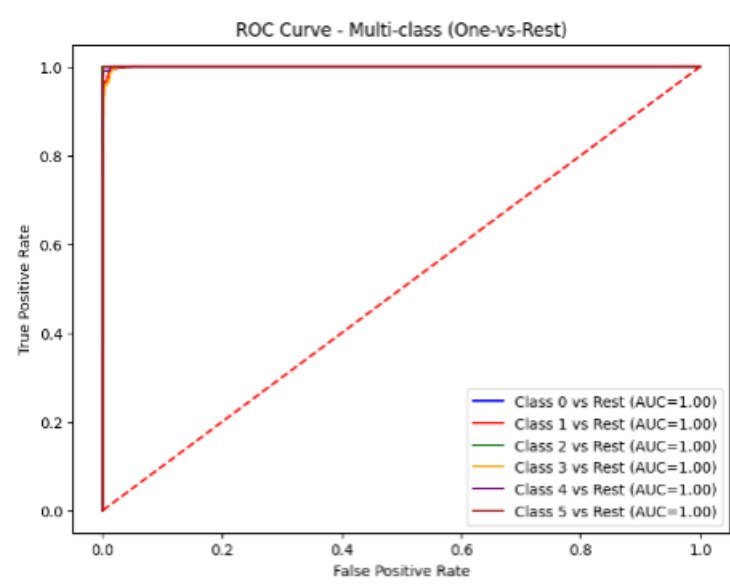
- Khi chạy Spark Streaming, chỉ việc load lại mô hình đã lưu này để dự đoán.



Hình 19: Confusion Matrix (số index)



Hình 20: Confusion Matrix (tên nhãn)



Hình 21: ROC Curve - Multi-class (One-vs-Rest)

```
model.write().overwrite().save("/content/drive/My Drive/BigData_BCOCK/rf_model_pipeline")
```

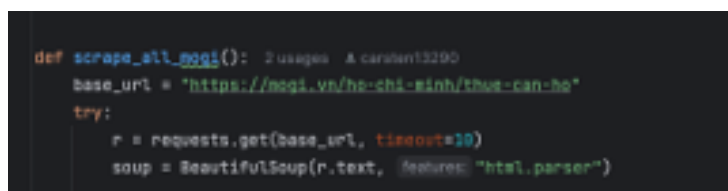
Hình 22: Lưu mô hình

THỰC NGHIỆM

Triển khai

1. Producer (web_scraper_kafka.py):

- Cào liên tục tin đăng Mogi, parse thành JSON, gửi vào topic “mogi-stream”.



```
def scrape_all_mogi():  
    base_url = "https://mogi.vn/hc-chi-minh/thue-can-ho"  
    try:  
        r = requests.get(base_url, timeout=10)  
        soup = BeautifulSoup(r.text, features="html.parser")
```

Hình 23: Cào liên tục tin đăng Mogi

2. Consumer (Spark Streaming):

- Đọc dữ liệu từ Kafka topic “mogi-stream”, parse JSON → DataFrame.
- Phương thức để xử lý dữ liệu (streaming trực tiếp từ website Mogi) lại một lần nữa để phù hợp với Model “rf_model_pipeline”.

Kết quả đánh giá

- Cào liên tục tin đăng Mogi, parse thành JSON, gửi vào topic “mogi-stream”.
- Tốc độ xử lý: Spark Structured Streaming xử lý micro-batch tầm 10 giây, đủ linh hoạt cho luồng tin mới.
- Load mô hình “rf_model_pipeline” (Random Forest Pipeline) đã huấn luyện.
- Gọi `model.transform(df)` để sinh prediction.
- Confusion matrix thể hiện khả năng phân biệt giữa các phân khúc thuê nhà.
- Ghi kết quả ra topic “mogi-predictions” hoặc in ra console.

Hạn chế

- Cần cấu hình kỹ khi dữ liệu quá lớn để tránh tắc nghẽn (bottleneck).
- Quá trình geocoding hoặc tính tiện ích xung quanh phụ thuộc vào API ngoài, dễ gặp giới hạn hoặc độ trễ.

```

def normalize_address(addr):

def geocode_location(location_str):

def distance_to_center(lat, lon):

def get_count_avgdist(lat, lon, amenity="hospital", radius=3000):

def query_osm_supermarket(lat, lon, radius_km=5):

def calculate_supermarket_stats(lat, lon, radius_km=5):

def parse_price(raw_price):

def scrape_all_mogi(): 2 usages ± carsten13290
    base_url = "https://mogi.vn/ho-chi-minh/thue-can-ho"
    try:
        r = requests.get(base_url, timeout=10)
        soup = BeautifulSoup(r.text, features="html.parser")

```

Hình 24: Tiếp tục tiền xử lý dữ liệu

```

data = {
    "price (million VND)": price_vnd,
    "square (m2)": sq,
    "bedroom": bd,
    "restroom": rr,
    "distance_to_center": dist_center,
    "hospital_count": hosp_count,
    "hospital_avg_distance": hosp_avg,
    "school_count": school_count,
    "school_avg_distance": school_avg,
    "super_market_count": market_count,
    "super_market_avg_distance": market_avg,
    "location": location,
    "date": date_,
    "url": url_
}

return data

```

Hình 25: Dữ liệu streaming được xử lý chính xác với yêu cầu của Model

```

hàng số tự nhiên: https://mogi.vn/guest-1/thu-can-ho-chung-cu-tan-cho-thu-can-to-1p-vietnam-gi-len-rim-ha-son-51a2-sang-trang-1a21a6979
lat/lon không có, sử dụng geocoding từ địa chỉ...
[PRODUCER] Sent data: {'price (million VND)': 2900.0, 'square (m2)': 20.0, 'bedroom': 3, 'restroom': 3, 'distance_to_center': 7.7019211621797, 'hospital_count':
5, 'hospital_avg_distance': 2.65612796134728, 'school_count': 15, 'school_avg_distance': 2.71792170050943, 'super_market_count': 25, 'super_market_avg_distance':
2.22205202629407, 'location': 'Nguyễn Lương Bằng, Phường Phú Mỹ, Quận 7, TP HCM', 'date': '17/02/2023', 'url': 'https://mogi.vn/guest-1/thu-can-ho-chung-cu-tan-cho-thu-can-to-1p-vietnam-gi-len-rim-ha-son-51a2-sang-trang-1a21a6979'}
lat/lon không có, sử dụng geocoding từ địa chỉ...
[PRODUCER] Sent data: {'price (million VND)': 3000.0, 'square (m2)': 20.0, 'bedroom': 3.0, 'restroom': 3.0, 'distance_to_center': 8.27714803443278, 'hospital_coun
nt': 15, 'hospital_avg_distance': 2.65612796134728, 'school_count': 15, 'school_avg_distance': 2.71792170050943, 'super_market_count': 25, 'super_market_avg_d
istance': 2.22205202629407, 'location': 'Nguyễn Văn Linh, Phường Tân Phong, Quận 7, TP HCM', 'date': '17/02/2023', 'url': 'https://mogi.vn/guest-1/thu-can-ho-chung-cu-tan-cho-thu-can-to-1p-vietnam-gi-len-rim-ha-son-51a2-sang-trang-1a21a6979'}

```

Hình 26: Cào liên tục tin đăng Mogi

```

Received: {'price (million VND)': 2800000.0, 'square (m2)': 70.0, 'bedroom': 2.0, 'restroom': 2.0, 'distance_to_center': 8.7061174318264, 'hospital_count': 23, 'hospital_avg
_distance': 3.49322409518574, 'school_count': 70, 'school_avg_distance': 3.484427902048056, 'super_market_count': 137, 'super_market_avg_distance': 2.76568095124135, 'locat
ion': 'Tân Mỹ Hưng, Phường Tân Mỹ, Quận 7, TP HCM', 'date': '17/02/2023', 'url': 'https://mogi.vn/guest-1/thu-can-ho-chung-cu-tan-cho-thu-can-to-1p-vietnam-gi-len-rim-ha-son-51a2-sang-trang-1a21a6979'}
[PRODUCER] Sent data: {'price (million VND)': 3200000.0, 'square (m2)': 80.0, 'bedroom': 2.0, 'restroom': 2.0, 'distance_to_center': 8.7061174318264, 'hospital_count': 23, 'hospital_avg
_distance': 3.49322409518574, 'school_count': 70, 'school_avg_distance': 3.484427902048056, 'super_market_count': 137, 'super_market_avg_distance': 2.76568095124135, 'locat
ion': 'Tân Mỹ Hưng, Phường Tân Mỹ, Quận 7, TP HCM', 'date': '17/02/2023', 'url': 'https://mogi.vn/guest-1/thu-can-ho-chung-cu-tan-cho-thu-can-to-1p-vietnam-gi-len-rim-ha-son-51a2-sang-trang-1a21a6979'}

```

Hình 27: Parse thành JSON, gửi vào topic “mogi-stream”

```

{"price (million VND)":square (m2):bedroom:restroom:distance_to_center:hospital_count:hospital_avg_distance:school_count:school_avg_distance:super_market_count:super_market_avg_d
istance:location}
dateurl
Features
probability
prediction
[{"price (million VND)": 2800000.0, "square (m2)": 70.0, "bedroom": 2.0, "restroom": 2.0, "distance_to_center": 8.7061174318264, "hospital_count": 23, "hospital_avg_distance": 3.49322409518574, "school_count": 70, "school_avg_distance": 3.484427902048056, "super_market_count": 137, "super_market_avg_distance": 2.76568095124135, "location": "Tân Mỹ Hưng, Phường Tân Mỹ, Quận 7, TP HCM", "date": "17/02/2023", "url": "https://mogi.vn/guest-1/thu-can-ho-chung-cu-tan-cho-thu-can-to-1p-vietnam-gi-len-rim-ha-son-51a2-sang-trang-1a21a6979"}, {"price (million VND)": 3200000.0, "square (m2)": 80.0, "bedroom": 2.0, "restroom": 2.0, "distance_to_center": 8.7061174318264, "hospital_count": 23, "hospital_avg_distance": 3.49322409518574, "school_count": 70, "school_avg_distance": 3.484427902048056, "super_market_count": 137, "super_market_avg_distance": 2.76568095124135, "location": "Tân Mỹ Hưng, Phường Tân Mỹ, Quận 7, TP HCM", "date": "17/02/2023", "url": "https://mogi.vn/guest-1/thu-can-ho-chung-cu-tan-cho-thu-can-to-1p-vietnam-gi-len-rim-ha-son-51a2-sang-trang-1a21a6979"}]

```

Hình 28: Spark lấy dữ liệu từ Kafka rồi load Model đã lưu lên xử lý

[illegible]

Hình 29: Kết quả sau khi Model xử lý data bản ghi của từng batch

DEMO

Chạy Producer

- Python `web_scraper_kafka.py`
- Mỗi 10 giây, sẽ gửi ~ 50 tin vào “mogi-stream”.

Chạy Spark Streaming

- `spark-submit spark_consumer_model.py`
- Đọc topic “mogi-stream”, xử lý, dự đoán, in ra console, gửi “mogi-predictions”.

Luồng kết quả

- Tin đăng A: giá 5 triệu, diện tích 30 m², cách trung tâm 5 km, ít tiện ích
→ Prediction: “người trẻ độc thân thu nhập trung bình”.
- Tin đăng B: giá 20 triệu, diện tích 80 m², gần trung tâm, nhiều tiện ích
→ Prediction: “gia đình thu nhập cao”.

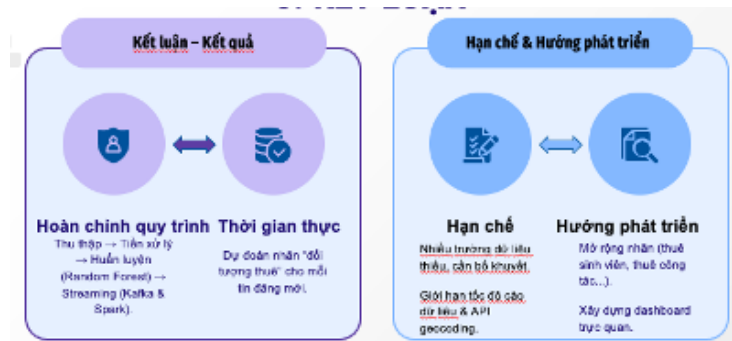
KẾT LUẬN

Kết quả

- Hoàn chỉnh quy trình thu thập dữ liệu (từ Mogi) – tiền xử lý – huấn luyện (Random Forest) – tích hợp streaming (Kafka & Spark).
- Xử lý thời gian thực: mỗi tin đăng mới sẽ tự động được dự đoán nhãn “đối tượng thuê”.

Hạn chế và hướng phát triển

- **Hạn chế:**
 - Dữ liệu thực tế có thể nhiều trường “mất” giá trị, cần áp dụng phương pháp bổ khuyết phức tạp hơn.
 - Giới hạn tốc độ cào dữ liệu và giới hạn API (geocoding, Overpass).
- **Hướng phát triển:**
 - Mở rộng nhãn phân khúc (ví dụ “nhóm thuê sinh viên”, “nhóm thuê công tác ngắn hạn”,...).
 - Triển khai mô hình online learning để tự động thích nghi khi thị trường thay đổi.
 - Xây dựng dashboard tương tác trực quan, cho phép quản trị viên theo dõi hiệu suất dự đoán.



Hình 30: Kết luận

TÀI LIỆU THAM KHẢO

- Apache Spark - Structured Streaming Programming Guide. [Trực tuyến]. Địa chỉ: <https://spark.apache.org/docs/latest/structured-streaming-programming-guide.html> [Truy cập lần cuối 17/02/2025]
- Kafka - Apache Kafka Documentation. [Trực tuyến]. Địa chỉ: <https://kafka.apache.org/documentation> [Truy cập lần cuối 17/02/2025]
- scikit-learn - Random Forest Documentation. [Trực tuyến]. Địa chỉ: <https://spark.apache.org/docs/latest/api/python/reference/pyspark.sql/functions.html> [Truy cập lần cuối 17/02/2025]
- Overpass API - OpenStreetMap Overpass Documentation. [Trực tuyến]. Địa chỉ: <https://overpass-api.de/> [Truy cập lần cuối 17/02/2025]
- Nominatim - Geopy Documentation. [Trực tuyến]. Địa chỉ: <https://geopy.readthedocs.io/en/stable/#nominatim> [Truy cập lần cuối 17/02/2025]