ĐẠI HỌC BÁCH KHOA HÀ NỘI

TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO BÀI TẬP LỚN AI

Chủ đề: Dự đoán giá nhà đất Hà Nội

Giảng viên hướng dẫn: **Trần Thế Hùng**

Nhóm 12

Sinh viên tham gia: **Phạm Công Hào 20215045**
Đặng Thái Tuấn 20210907
Phạm Đức Lưu 20215084

Hà Nội, 6/2024

I) Giới thiệu bài toán

- Bài toán dự đoán giá nhà đất là một trong những bài toán quan trọng trong lĩnh vực bất động sản. Mục đích của bài toán là xây dựng một mô hình dự đoán giá nhà đất dựa trên các yếu tố đầu vào như vị trí, diện tích, số phòng, số tầng và nhiều yếu tố khác. Với sự phát triển không ngừng và nhu cầu nhà ở tăng cao, việc dự đoán chính xác giá nhà đất sẽ giúp các nhà đầu tư, người mua nhà và các bên liên quan đưa ra các quyết định chính xác và hiệu quả hơn.

II) Phân công công việc

Công việc	**Tham gia**
Craw data	Đặng Thái Tuấn, Phạm Đức Lưu
Xử lý dữ liệu	Phạm Công Hào
Xây dựng mô hình	Phạm Công Hào, Đặng Thái Tuấn, Phạm Đức Lưu
Xây dựng giao diện	Phạm Đức Lưu, Đặng Thái Tuấn

III) Chi tiết công việc

a. Crawl data

Data được lấy từ page: [Giá Nhà Đất TPHCM Và Hà Nội Cập Nhật Mới Nhất T6/2024 (mogi.vn)]
Số lượng data Nhà: 55271
Số lượng data Chung cư: 1567

Phương pháp để crawl data:

Selenium: truy cập các phần tử của trang web để lấy dữ liệu trên các trang.

b. Xử lý dữ liệu

1. Missing values (xử lý các bản ghi bị thiếu dữ liệu)
   - Sau khi kiểm tra lượng dữ liệu bị thiếu trong mỗi bản ghi là 3% < 10% => thực hiện sinh dữ liệu cho các feature bị thiếu. Cụ thể:

     - Diện tích: gán giá trị bằng giá trị trung bình diện tích của các căn nhà trong bộ dữ liệu.
     - Số tầng: gán giá trị bằng giá trị xuất hiện nhiều lần nhất trong Feature “Số tầng” của bộ dữ liệu.
     - Số phòng ngủ, WC, Thang máy: Xử lý tương tự như “Số tầng”.
     - Mặt tiền: việc gán giá trị bị thiếu cho Feature này tùy thuộc vào vị trí của căn nhà trong bộ dữ liệu:
       - “Nhà hẻm, ngõ” => bằng 0.

        - “Nhà mặt tiền, phố” => bằng 5.
        - “Biệt Thự liền kề” => bằng giá trị trung bình cộng Mặt tiền của các căn nhà có vị trí “Biệt thự liền kề”.
    - Vị trí, Quận, Pháp lý: gán các giá trị bằng giá trị xuất hiện nhiều nhất trong mỗi Feature tương ứng.
2. Loại bỏ các Outliers (xử lý các bản ghi có dữ liệu khác nhiều so với đa số)
    - Sử dụng phương pháp Interquartile Range (IQR) tính giới hạn trên và giới hạn dưới sau đó loại bỏ các bản ghi có giá trị nằm ngoài khoảng từ giới hạn dưới đến giới hạn trên.
    - Ghi chú: bản ghi là các thông số của một căn nhà (diện tích, số tầng, vị trí, ...). Bộ dữ liệu được tạo thành từ các bản ghi.
3. Biến đổi dữ liệu dạng chữ
   Do các Feature bên dưới đều là kiểu dữ liệu dạng Ordinal feature (có thứ tự) nên cần mã hóa phân cấp cho từng Feature. Cụ thể:
    - Quận: gán bằng giá trị cho 1m2 đất của mỗi Quận. Cụ thể:
        - Quận Đống Đa: 192
        - Quận Hai Bà Trưng: 214
        - Quận Hà Đông: 144

        - ....
    - Vị trí: có các loại giá trị (“Nhà hẻm, ngõ”, “nan”, “Nhà đường nội bộ, Cổ Nhuế”, “Biệt thự liền kề”, “Nhà mặt tiền, phố”) sẽ được mã hóa lần lượt (0, 1, 1, 2, 3).

      Giải thích giá trị “nan” do dữ liệu của Chung cư không có thuộc tính “Vị trí” => không xác định => gán giá trị “nan”.
    - Pháp lý: có các giá trị (“Sổ hồng, Sổ đỏ”) do chỉ có hai thuộc tính để tránh việc mô hình hiểu nhầm đây là dữ liệu kiểu Boolean sẽ mã hóa (0, 2) thay vì (0, 1).
    - Kiểu nhà: có hai giá trị (“Nhà”, Chung cư”) tương tự sẽ được mã hóa (0, 2).

c. Xây dựng mô hình

   Giới thiệu: phương pháp Ensemble là một kỹ thuật trong học máy nhằm cải thiện độ chính xác và hiệu suất của mô hình bằng cách kết hợp nhiều mô hình dự đoán khác nhau thay vì chỉ dựa vào một mô hình duy nhất, phương pháp Ensemble tận dụng sức mạnh của nhiều mô hình để tạo ra một mô hình tổng hợp mạnh mẽ hơn.

   Nhóm sử dụng phương pháp Ensemble để kết hợp hai mô hình dự đoán Random Forest Regressor và XGBoost

   1. RandomForestRegressor (RFR)

- Cách thức hoạt động: Xây dựng n Decision Tree mỗi cây được train trên 1 bộ dữ liệu được lấy random từ bộ train. Vd 70% (có thể về số lượng bản ghi hoặc số lượng feature) mỗi Decision Tree sẽ học và đưa ra một giá trị dự đoán. Trung bình các kết quả dự đoán sẽ được lấy làm predict của model.
- Cách RFR xây dựng các cây con:
  B1: Khởi tạo: các node của cây được chọn ngẫu nhiên từ các Feature của bộ dữ liệu.
  B2: tính toán hàm mất mát (criterion) trên từng cây so sánh kết quả để chọn ra node root cho cây.
  B3: lặp lại B1 để xây dựng các node bên dưới của cây.
  Các bước trên được lặp đi lặp lại cho đến khi đạt được điều kiện dừng (kết quả của criterion không thay đổi, độ sâu tối đa của cây, số lượng mẫu tối thiểu tại mỗi node)
- Các tham số được sử dụng cho mô hình Random ForesstRegressor
  - n_estimators = 200 :số lượng cây con được tạo ra
  - criterion = “absolute_error” :mô hình sử dụng sai số tuyệt đối trung bình (MAE) để đo lượng chất lượng phân chia.
  - max_depth = 10: độ sâu tối đa của mỗi cây là 10.

2. XGBoost

- Khác với Random Forest Regressor, XGBoost xây dựng theo phương pháp boosting thay vì bagging. Các cây được xây dựng tuần tự, mỗi cây mới được xây dựng để giảm thiểu lỗi của cây hiện tại thông qua việc học hỏi lại kinh nghiệm của cây trước đó.
- Trong quá trình xây dựng tự động cắt tỉa cây: loại bỏ những nodes, leaves không đem lại giá trị tích cực trong quá trình mở rộng cây.

3. Các thông số về hiệu năng của model

```
R2: 0.6983178201309745
MSE: 1.6737251388446668
MAE: 0.7136409096365847
```

- R2: (càng gần 1 càng tốt)
- MSE: sai số tuyệt đối trung bình bậc 2 (càng thấp càng tốt)
- MAE: sai số tuyệt đối trung bình bậc 1(càng thấp càng tốt)

d. Xây dựng giao diện

Giao diện được xây dựng dựa trên thư viện tkinter