python 中的数据处理

杨晨
学号 2021212171
北京邮电大学计算机学院

日期：2024 年 2 月 26 日

1 概述

1.1 实验内容

1. 对你爬取下来的北京二手房数据，进行数据的预处理，并计算：
   (a) 四个区的平均总价、最高总价、最低总价；
   (b) 四个区的平均单价、最高单价、最低单价；
   (c) 按照房屋建成的年份，计算 2000 年以前、2000-2009.12.31、2010-至今，这三个时间段的平均单价。
2. 处理北京空气质量数据
   对 HUMI、PRES、TEMP 三列，进行线性插值处理。修改 cbwd 列中值为“cv”的单元格，其值用后项数据填充。

1.2 开发环境

- Windows10
- PyCharm 2023.2.4 (Professional Edition)

2 实验过程

2.1 统计二手房数据

2.1.1 概述

由于之前爬取的数据里，没有房屋的建成年份，所以修改`spider.py`的代码，重新爬取

```
def parse(self, response):
    item = LianjiaItem()
    distinct = response.url.split("/")[4]
    page = response.url.split("/")[5]
    for each in response.xpath('//ul[@class="sellListContent"]/li'):
        item["name"] = each.xpath("div/div/a/text()").get()
        price_value = each.xpath(
```

```
            "div/div[@class='priceInfo']/div[@class='totalPrice totalPrice2']/span/
                text()"
        ).get()
        price_unit = each.xpath(
            "div/div[@class='priceInfo']/div[@class='totalPrice totalPrice2']/i[
                last()]/text()"
        ).get()
        item["price"] = f"{price_value}{price_unit}"
        area_text = each.xpath(
            ".//div[@class='address']/div[@class='houseInfo']/text()"
        ).get()
        square = re.search(r"(\d+(\.\d+)?)平米", area_text)
        build_year = re.search(r"(\d+)年", area_text)
        if square:
            item["area"] = square.group(1) + "平米"
        else:
            item["area"] = ""
        if build_year:
            item["built_time"] = build_year.group(1) + "年"
        else:
            item["built_time"] = ""
        item["unit_price"] = each.xpath(
            "div/div[@class='priceInfo']/div[@class='unitPrice']/span/text()"
        ).get()
        item["distinct"] = distinct
        yield item
```

LianjiaData.json爬取结果如下，这里展示部分


```
{"name": "北苑满五年唯一，南北通透板楼，双卫，双阳台，高层", "price": "815万",
    "area": "153.47平米", "built_time": "", "unit_price": "53,105元/平",
    "distinct": "chaoyang"}
{"name": "望京西园三区新上，南北四居，阳台俯瞰小区花园全景。", "price": "1200万",
    "area": "218.17平米", "built_time": "2001年", "unit_price": "55,003元/平",
    "distinct": "chaoyang"}
{"name": "峻峰华亭 跃层灵动格局 私密性好 停车位充足 四环内", "price": "820万",
    "area": "161.68平米", "built_time": "", "unit_price": "50,718元/平",
    "distinct": "chaoyang"}
{"name": "满五唯一。无抵押户口，临河明卫，南北三居，诚意出售", "price": "755万",
    "area": "150.12平米", "built_time": "", "unit_price": "50,294元/平",
    "distinct": "chaoyang"}
{"name": "柳芳南里 3室1厅 南 北", "price": "760万", "area": "88.77平米",
    "built_time": "1990年", "unit_price": "85,615元/平", "distinct": "chaoyang"}
{"name": "和乔丽晶一期 低密度板楼 中间楼层诚意出售", "price": "990万", "area":
    "167.66平米", "built_time": "2000年", "unit_price": "59,049元/平", "distinct":
    "chaoyang"}
{"name": "满五唯一，南北通透 错层三居，不临街 采光好", "price": "730万", "area":
    "172.17平米", "built_time": "", "unit_price": "42,400元/平", "distinct":
```

```
    "chaoyang"}
{"name": "中直小区，管理不错，全明格局，交通便利，配套齐全", "price": "720万",
    "area": "72.22平米", "built_time": "", "unit_price": "99,696元/平", "distinct":
    "chaoyang"}
......
```

2.1.2 统计过程

根据 json 文件中的内容，可以发现，房屋的区域、总价和单价数据都是齐全的，而建成年份只有部分房屋有。所以在用正则表达式提取时，需要注意匹配结果。

```
for item in data:
    distinct = item["distinct"]
    price = re.search(r"(\d+(\.\d+)?)万", item["price"]).group(1)
    unit = re.search(r"(\d+(,\d+)?)元/平", item["unit_price"]).group(1).replace(","
        , "")
    total_price[distinct].append(float(price))
    unit_price[distinct].append(float(unit))
    year = re.search(r"(\d+)年", item["built_time"])
    if year:  # 有些房源没有建造时间，所以需要判断
        year = int(year.group(1))
        if year < 2000:
            built_years["before_2000"].append(float(unit))
        elif year < 2010:
            built_years["2000_to_2009"].append(float(unit))
        else:
            built_years["2010_and_after"].append(float(unit))
```

之后，按照区域，统计平均总价、最高总价、最低总价；统计平均单价、最高单价、最低单价

```
for distinct in total_price:
    print(f"{distinct}区二手房总价最高的房源：{max(total_price[distinct]):}万")
    print(f"{distinct}区二手房总价最低的房源：{min(total_price[distinct]):}万")
    print(f"{distinct}区二手房总价均值：{sum(total_price[distinct])/len(total_price
        [distinct]):.2f}万\n")
    print(f"{distinct}区二手房单价最高的房源：{max(unit_price[distinct]):}元/平米")
    print(f"{distinct}区二手房单价最低的房源：{min(unit_price[distinct]):}元/平米")
    print(f"{distinct}区二手房单价均值：{sum(unit_price[distinct])/len(unit_price[
        distinct]):.2f}元/平米\n")
```

之后，对于所有有建造年份的房屋，按照时间段，统计平均单价

```
for year_range in built_years:
    print(f"{year_range}建造的二手房单价均值：{sum(built_years[year_range])/len(
        built_years[year_range]):.2f}元/平米")
```

2.1.3 统计结果

```
dongcheng区二手房总价最高的房源：2600.0万
dongcheng区二手房总价最低的房源：330.0万
dongcheng区二手房总价均值：761.52万

dongcheng区二手房单价最高的房源：155669.0元/平米
dongcheng区二手房单价最低的房源：31172.0元/平米
dongcheng区二手房单价均值：102411.30元/平米

xicheng区二手房总价最高的房源：3125.0万
xicheng区二手房总价最低的房源：349.0万
xicheng区二手房总价均值：888.65万

xicheng区二手房单价最高的房源：179731.0元/平米
xicheng区二手房单价最低的房源：38937.0元/平米
xicheng区二手房单价均值：118588.37元/平米

chaoyang区二手房总价最高的房源：2980.0万
chaoyang区二手房总价最低的房源：140.0万
chaoyang区二手房总价均值：616.99万

chaoyang区二手房单价最高的房源：104247.0元/平米
chaoyang区二手房单价最低的房源：33760.0元/平米
chaoyang区二手房单价均值：65578.25元/平米

haidian区二手房总价最高的房源：2799.0万
haidian区二手房总价最低的房源：253.0万
haidian区二手房总价均值：817.13万

haidian区二手房单价最高的房源：152812.0元/平米
haidian区二手房单价最低的房源：32971.0元/平米
haidian区二手房单价均值：92072.74元/平米

before_2000建造的二手房单价均值：94497.74元/平米
2000_to_2009建造的二手房单价均值：95625.52元/平米
2010_and_after建造的二手房单价均值：81764.13元/平米
```

2.2 csv 文件填充

1. 导入所需的库

   ```
   import pandas as pd
   ```

   导入了 pandas 库，用于数据处理和分析。

2. 读取 CSV 文件

```
data = pd.read_csv(
    r"C:\Users\Administrator\Documents\Tencent Files\1369792882\FileRecv\
        BeijingPM20100101_20151231.csv",
    encoding="utf-8"
)
```

使用pd.read_csv函数读取 CSV 文件。提供了文件路径，并指定了编码为 UTF-8。

3. 线性插值处理

```
data["HUMI"] = data["HUMI"].interpolate()
data["PRES"] = data["PRES"].interpolate()
data["TEMP"] = data["TEMP"].interpolate()
```

对”HUMI”、”PRES” 和”TEMP” 三列进行线性插值处理。使用interpolate函数填充这些列中的缺失值，根据前后数据的趋势进行插值。

4. 修改特定列的值

```
data["cbwd"] = data["cbwd"].replace("cv", method="bfill")
```

将”cbwd” 列中值为”cv” 的单元格用后项数据进行填充。使用replace函数替换特定值。

5. 保存处理后的数据

```
data.to_csv("processed.csv", index=False, encoding="utf-8")
```

使用to_csv函数将处理后的数据保存为名为”processed.csv” 的新文件。设置index=False参数以不保存索引列，并指定编码为 UTF-8。

通过运行以上代码，将执行以下操作：

- 加载 CSV 文件”BeijingPM20100101_20151231.csv”。
- 对”HUMI”、”PRES” 和”TEMP” 三列进行线性插值处理，填充缺失值。
- 修改”cbwd” 列中值为”cv” 的单元格，用后项数据进行填充。
- 将处理后的数据保存为名为”processed.csv” 的新文件。

3 附录：完整代码

3.1 统计二手房数据

```
import json
import re

data = []

with open(r"../py_homework/LianjiaData.json", "r", encoding="utf-8") as f:
    for line in f:
        item = json.loads(line)
        data.append(item)
```

```
total_price = {"dongcheng": [], "xicheng": [], "chaoyang": [], "haidian": []}
unit_price = {"dongcheng": [], "xicheng": [], "chaoyang": [], "haidian": []}
built_years = {"before_2000": [], "2000_to_2009": [], "2010_and_after": []}

for item in data:
    distinct = item["distinct"]
    price = re.search(r"(\d+(\.\d+)?)万", item["price"]).group(1)
    unit = re.search(r"(\d+(,\d+)?)元/平", item["unit_price"]).group(1).replace(","
        , "")
    total_price[distinct].append(float(price))
    unit_price[distinct].append(float(unit))
    year = re.search(r"(\d+)年", item["built_time"])
    if year:  # 有些房源没有建造时间, 所以需要判断
        year = int(year.group(1))
        if year < 2000:
            built_years["before_2000"].append(float(unit))
        elif year < 2010:
            built_years["2000_to_2009"].append(float(unit))
        else:
            built_years["2010_and_after"].append(float(unit))


for distinct in total_price:
    print(f"{distinct}区二手房总价最高的房源: {max(total_price[distinct]):}万")
    print(f"{distinct}区二手房总价最低的房源: {min(total_price[distinct]):}万")
    print(f"{distinct}区二手房总价均值: {sum(total_price[distinct])/len(total_price
        [distinct]):.2f}万\n")
    print(f"{distinct}区二手房单价最高的房源: {max(unit_price[distinct]):}元/平米")
    print(f"{distinct}区二手房单价最低的房源: {min(unit_price[distinct]):}元/平米")
    print(f"{distinct}区二手房单价均值: {sum(unit_price[distinct])/len(unit_price[
        distinct]):.2f}元/平米\n")

for year_range in built_years:
    print(f"{year_range}建造的二手房单价均值: {sum(built_years[year_range])/len(
        built_years[year_range]):.2f}元/平米")
```

3.2 csv 文件填充

```
import pandas as pd

data = pd.read_csv(
    r"C:\Users\Administrator\Documents\Tencent Files\1369792882\FileRecv\
        BeijingPM20100101_20151231.csv",
    encoding="utf-8",
)
```

```
# 对HUMI、PRES、TEMP三列，进行线性插值处理
data["HUMI"] = data["HUMI"].interpolate()
data["PRES"] = data["PRES"].interpolate()
data["TEMP"] = data["TEMP"].interpolate()

# 修改cbwd列中值为“cv”的单元格，其值用后项数据填充
data["cbwd"] = data["cbwd"].replace("cv", method="bfill")

data.to_csv("processed.csv", index=False, encoding="utf-8")
```