Housing Price Analysis and Prediction

1조

Individual role

김연수

Data Analysis

배건우

Web Service

최성원

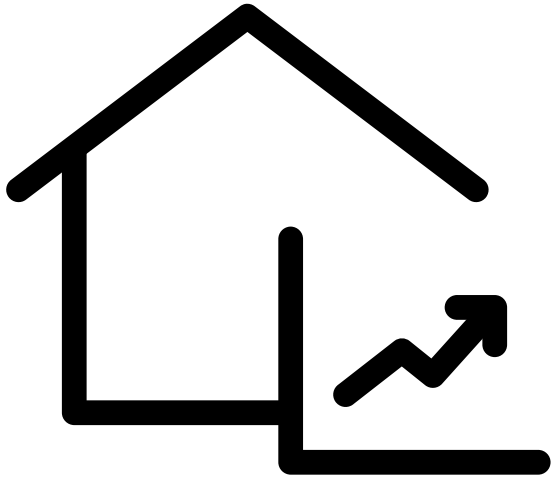
Data Analysis

최예리

Web Service

CONTENTS

Overview



Analysis and Predict Housing Price by Real Estate Data

Goal

Analysis

- Compare Housing Price
- Analyze Housing Price Change

ML

- Predict Housing Price

Visualization

- Visualize analysis and prediction result

Web Service

- Users can search housing price
- Real estate news tab

Approach

1. Acquisition and Data preprocessing

1) Real Estate Data

- Acquire data from “국토교통부 부동산 실거래가 공개시스템”
- Data preprocessing by Pandas (Data Size : 657.5 MB)

Approach

1. Acquisition and Data preprocessing

2) Real Estate News

Crawling real estate news data (Data size : 124 KB)

Approach

2. Storage

```
hadoop@master:hadoop-2.7.6/bin $ ./hadoop dfs -ls ./dataset
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

Found 17 items
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:09 dataset/강원도
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:07 dataset/경기도
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:07 dataset/경상남도
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:07 dataset/경상북도
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:07 dataset/광주광역시
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:07 dataset/대구광역시
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:07 dataset/대전광역시
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:07 dataset/부산광역시
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:07 dataset/서울특별시
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:07 dataset/세종특별자치시
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:07 dataset/울산광역시
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:08 dataset/인천광역시
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:08 dataset/전라남도
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:08 dataset/전라북도
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:08 dataset/제주특별자치도
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:08 dataset/충청남도
drwxr-xr-x   - hadoop supergroup          0 2021-05-24 01:08 dataset/충청북도
```

Clean data and store it in HDFS

Approach

3. Analysis

Analysis and Predict a Housing price using Spark

Approach

4. Visualization

Data Extraction using SparkSQL
and Visualization using Zeppelin

Approach

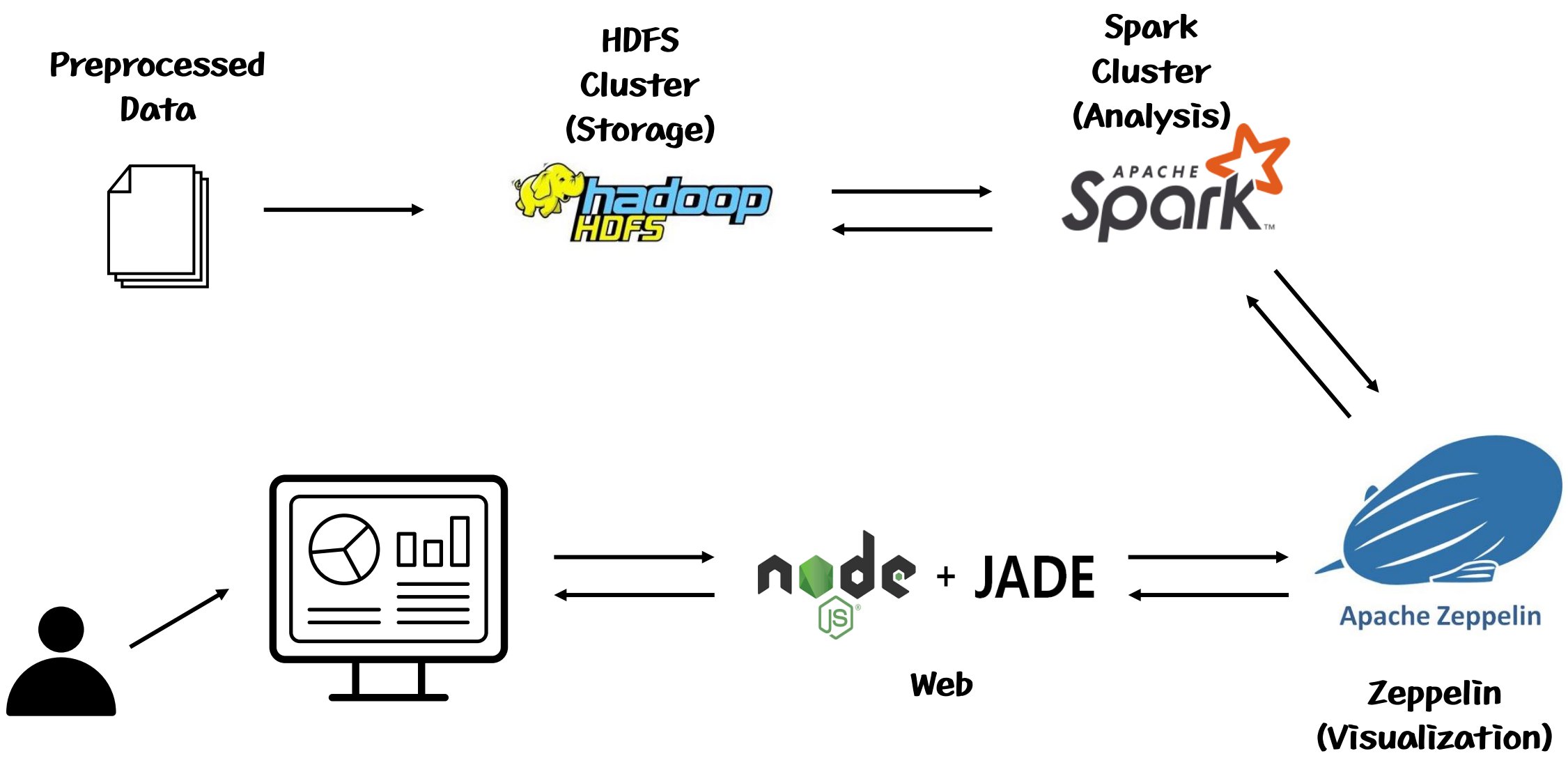
5. Web

Backend : node.js, JavaScript
Frontend : JADE, CSS template engine

Development Environment

- Hadoop 2.7.6
- Spark 2.2.2
- Zeppelin 0.8.2
- Python 3.7
- Nodejs v5.12.0
- Npm 3.8.6

Architecture



Code

```
%pyspark
basic_folder = "hdfs://master:9000/user/hadoop/dataset/"

provinces = ["강원도","경기도","경상남도","경상북도","광주광역시",
             "대구광역시","대전광역시","부산광역시","서울특별시",
             "세종특별자치시","울산광역시","인천광역시","전라남도",
             "전라북도","제주특별자치도","충청남도","충청북도"]
```

```
%pyspark

for p in provinces:
    files = basic_folder + p + '/' + p + '_*'
    print(files)
    df = spark.read.csv(files, header=True, inferSchema=True)
    df.createOrReplaceTempView('`' + p + '`')
```

Load data from HDFS and Create TempView by province

Code

```
%pyspark
basic_folder = "hdfs://master:9000/user/hadoop/

provinces = ["강원도","경기도","경상남도","경상
             "대구광역시","대전광역시","부산광
             "세종특별자치시","울산광역시","인
             "전라북도","제주특별자치도","충청남
```

```
%pyspark

for p in provinces:
    files = basic_folder + p + '/' + p + '_*'
    print(files)
    df = spark.read.csv(files, header=True, i
    df.createOrReplaceTempView('`' + p + '`')
```

```
hdfs://master:9000/user/hadoop/dataset/강원도/강원도_*
hdfs://master:9000/user/hadoop/dataset/경기도/경기도_*
hdfs://master:9000/user/hadoop/dataset/경상남도/경상남도_*
hdfs://master:9000/user/hadoop/dataset/경상북도/경상북도_*
hdfs://master:9000/user/hadoop/dataset/광주광역시/광주광역시_*
hdfs://master:9000/user/hadoop/dataset/대구광역시/대구광역시_*
hdfs://master:9000/user/hadoop/dataset/대전광역시/대전광역시_*
hdfs://master:9000/user/hadoop/dataset/부산광역시/부산광역시_*
hdfs://master:9000/user/hadoop/dataset/서울특별시/서울특별시_*
hdfs://master:9000/user/hadoop/dataset/세종특별자치시/세종특별자치시_*
hdfs://master:9000/user/hadoop/dataset/울산광역시/울산광역시_*
hdfs://master:9000/user/hadoop/dataset/인천광역시/인천광역시_*
hdfs://master:9000/user/hadoop/dataset/전라남도/전라남도_*
hdfs://master:9000/user/hadoop/dataset/전라북도/전라북도_*
hdfs://master:9000/user/hadoop/dataset/제주특별자치도/제주특별자치도_*
hdfs://master:9000/user/hadoop/dataset/충청남도/충청남도_*
hdfs://master:9000/user/hadoop/dataset/충청북도/충청북도_*
```

Load data from HDFS and Create TempView by province

Code

```
%pyspark
# 시도별 검색
z.put("province",z.input("특별시/도"))
z.z.run("20210602-223915_1629862033")
```

```
%pyspark
# 시군구 검색
z.put("detailAddress", z.input("시군구"))
z.z.run("20210608-223417_338745986")
```

```
%pyspark
# 아파트별 검색, 아파트 시세 예측
z.put("apart",z.input("아파트"))
z.z.run("20210604-205615_1591325726") # 아파트 차트
z.z.run("20210608-223722_241841938") # 아파트값 예측
```

```
%pyspark
# 집값에 따른 검색
z.put("price_up", z.input("집값(이상)"))
z.put("price_down", z.input("집값(이하)"))
z.z.run("20210608-223448_1865676816")
```

Get user input from Zeppelin dynamic form
and Trigger related paragraphs

Code

```
%pyspark
# 특별시/도별 차트
df_province = sqlContext.sql("""
select substr(`계약년월`, 1, 4) as y,
       cast(avg(regexp_replace(`거래금액(만원)`, ',', '')) as decimal) as price
from `""" + z.get("province") + """`
group by y
order by y
""")

z.show(df_province)
```

```
%pyspark
# 시군구별 차트
df_detailAddress = sqlContext.sql("""
select substr(`계약년월`, 1, 4) as year,
       cast(avg(regexp_replace(`거래금액(만원)`, ',', '')) as decimal) as price
from `""" + z.get("province") + """`
where `시군구` like '%""" + z.get("detailAddress") + """%'
group by year
order by year
""")

z.show(df_detailAddress)
```

```
%pyspark
# 아파트별 차트
df_apart = sqlContext.sql("""
select substr(`계약년월`, 1, 4) as y,
       cast(avg(regexp_replace(`거래금액(만원)`, ',', '')) as decimal) as price
from `""" + z.get("province") + """`
where `단지명` like '""" + z.get("apart") + """%'
group by y
order by y
""")

z.show(df_apart)
```

Extract data using SparkSQL and Create a Graph

Code

```
%pyspark
# 원하는 집값에 대한 데이터
df_recent = sqlContext.sql("""
select `단지명` as apt, max(`계약년월`) as `계약년월`,
       max(cast(regexp_replace(`거래금액(만원)`, ',', '') as decimal)) as price,
       round(`전용면적(㎡)`, 2), max(`건축년도`)
from `""" + z.get("province") + """`
where cast(regexp_replace(`거래금액(만원)`, ',', '') as decimal)
      between """ + z.get("price_up") + """ and """ + z.get("price_down") + """
      and (`시군구` like '%""" + z.get("detailAddress") + """%') and `계약년월` like '2021%'
group by apt, `전용면적(㎡)`
order by `계약년월` desc
""")

z.show(df_recent)
```

```
%pyspark
# 검색한 아파트의 층별 집값
df_floor = sqlContext.sql("""
select `층` as floor,
       cast(avg(regexp_replace(`거래금액(만원)`, ',', '')) as decimal) as price
from `""" + z.get("province") + """`
where (`단지명` like '""" + z.get("apart") + """%')
group by floor
order by floor
""")

z.show(df_floor)
```

Extract data using SparkSQL and Create a Graph

Code

```
%pyspark

df_apart_recent = sqlContext.sql("""
select `계약년월` as y,
       cast(avg(regexp_replace(`거래금액(만원)`, ',', '')) 
from `""" + z.get("province") + """`
where `단지명` like '""" + z.get("apart") + """%'
group by y
order by y desc limit 6
""")

z.z.run("20210605-175407_81665233") # 아파트값 예측 실행
```

```
%pyspark
df_apart.printSchema()
```

```
root
 |-- y: string (nullable = true)
 |-- price: decimal(10,0) (nullable = true)
```

```
%pyspark
# 아파트 검색 -> 예측값 반환
# get next year for pred_dataset
df_apart_recent.registerTempTable("temp")
df_pred_y = sqlContext.sql("""
select max(y) + 1 as y
from temp
""")

# trainset
from pyspark.sql.types import DateType
df_apart_recent = df_apart_recent.withColumn("y", df_apart_recent['y'].cast('int'))

from pyspark.ml.feature import VectorAssembler
feature_columns = df_apart_recent.columns[:-1]
assembler = VectorAssembler(inputCols=feature_columns,outputCol="year")
dataset = assembler.transform(df_apart_recent)
train, test = dataset.randomSplit([1.0, 0.0]) # train 100% of dataset

# model and train
from pyspark.ml.regression import LinearRegression
algo = LinearRegression(featuresCol="year", labelCol="price")
model = algo.fit(train)

# prediction
testset = assembler.transform(df_pred_y)
predictions = model.transform(testset)

# get predicted val
import pyspark.sql.functions as f
pred_val = predictions.select(f.collect_list('prediction')).first()[0][0]
print(pred_val)
```

Predict a housing price using SparkML

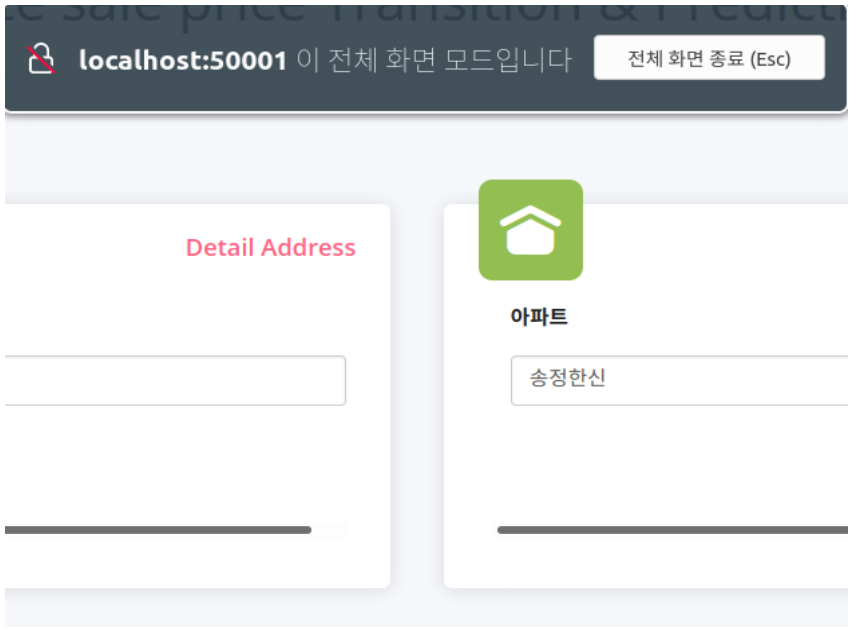
Code

```
doctype html
html(lang='en')
  head
    title
    // Meta
    meta(charset='utf-8')
    meta(name='viewport' content='width=device-width, initial-scale=1.0, user-scalable=0, minimal-ui')
    meta(http-equiv='X-UA-Compatible' content='IE=edge')
    meta(name='description' content='CodedThemes')
    meta(name='keywords' content=' Admin , Responsive, Landing, Bootstrap, App, Template, Mobile, iOS, Android, apple, creative app')
    meta(name='author' content='CodedThemes')
    // Favicon icon
    link(rel='icon' href='/assets/images/favicon.ico' type='image/x-icon')
    // Google font
    link(href='https://fonts.googleapis.com/css?family=Open+Sans:400,600' rel='stylesheet')
    // Required Fremwork
    link(rel='stylesheet' type='text/css' href='/assets/css/bootstrap/css/bootstrap.min.css')
    // themify-icons line icon
    link(rel='stylesheet' type='text/css' href='/assets/icon/themify-icons/themify-icons.css')
    // ico font
    link(rel='stylesheet' type='text/css' href='/assets/icon/icofont/css/icofont.css')
    // Style.css
    link(rel='stylesheet' type='text/css' href='/assets/css/style.css')
    link(rel='stylesheet' type='text/css' href='/assets/css/jquery.mCustomScrollbar.css')
  body
    // Pre-loader start
    .theme-loader
      .ball-scale
        .contain
          .ring
            .frame
          .ring
            .frame
          .ring
            .frame
          .ring
            .frame
          .ring
            .frame
          .ring
            .frame
          .ring
            .frame
          .ring
            .frame
          .ring
```

Link the css file and Preload

Code

```
// Pre-loader end
#pcoded.pcoded
  .pcoded-overlay-box
  .pcoded-container.navbar-wrapper
    nav.navbar.header-navbar.pcoded-header
      .navbar-wrapper
        .navbar-logo
          a(href='/' style='padding-left:10px;')
            h2 TEAM 1
        .navbar-container.container-fluid
          ul.nav-left
            li
              .sidebar_toggle
                a(href='javascript:void(0)')
                  i.ti-menu
            li
              a(href='#!' onclick='javascript:toggleFullScreen()')
                i.ti-fullscreen
          ul.nav-left
            li
              h2(style='padding-top:20px')
                | Real estate sale price Transition & Prediction
```



TEAM 1 ⤫ Real estate sale price Transition & Prediction

Make navigation bar and full-screen toggle button

Code

```
.col-md-6.col-xl-3
  .card.widget-card-1
    .card-block-small
      i.icofont.icofont-world.bg-c-blue.card1-icon
      span.text-c-blue.f-w-600 Province
      a.input
        iframe(src='http://133.186.244.163:50077/#/notebook/2G84Y3CMG/paragraph/20210605-154634_241069600?asIframe', frameborder='0')
      div
        span.f-left.m-t-10.text-muted
// card1 end
// card1 start
.col-md-6.col-xl-3
  .card.widget-card-1
    .card-block-small
      i.icofont.icofont-building.bg-c-pink.card1-icon
      span.text-c-pink.f-w-600 Detail Address
      a.input
        iframe(src='http://133.186.244.163:50077/#/notebook/2G84Y3CMG/paragraph/20210608-223940_184624074?asIframe', frameborder='0')
      div
        span.f-left.m-t-10.text-muted
// card1 end
// card1 start
.col-md-6.col-xl-3
  .card.widget-card-1
    .card-block-small
      i.icofont.icofont-ui-home.bg-c-green.card1-icon
      span.text-c-green.f-w-600 Apart
      a.input
        iframe(src='http://133.186.244.163:50077/#/notebook/2G84Y3CMG/paragraph/20210605-165454_942415786?asIframe', frameborder='0')
      div
        span.f-left.m-t-10.text-muted
// card1 end
// card1 start
.col-md-6.col-xl-3
  .card.widget-card-1
    .card-block-small
      i.icofont.icofont-cur-dollar.bg-c-yellow.card1-icon
      span.text-c-yellow.f-w-600 Price
      a.input
        iframe(src='http://133.186.244.163:50077/#/notebook/2G84Y3CMG/paragraph/20210608-230516_1100001444?asIframe', frameborder='0')
      div
        span.f-left.m-t-10.text-muted
          i.text-c-yellow.f-16.icofont.icofont-tag.m-r-10
          | Input price range
```

Make input box by using iframe

Code

```
// card1 end
// Statestics Start
.col-md-12.col-xl-12
  .card
    .card-header
      h5 Province Transition
      .card-header-left
      .card-header-right
        ul.list-unstyled.card-option
          li
            i.icofont.icofont-simple-left
          li
            i.icofont.icofont-maximize.full-card
          li
            i.icofont.icofont-minus.minimize-card
          li
            i.icofont.icofont-refresh.reload-card
    .card-block
      a.chart
        iframe(src='http://133.186.244.163:50077/#/notebook/2G84Y3CMG/paragraph/20210602-223915_1629862033?asIframe', frameborder='0', allowfullscreen)
.col-md-12.col-xl-12
  .card
    .card-header
      h5 Detail Address Transition
      .card-header-left
      .card-header-right
        ul.list-unstyled.card-option
          li
            i.icofont.icofont-simple-left
          li
            i.icofont.icofont-maximize.full-card
          li
            i.icofont.icofont-minus.minimize-card
          li
            i.icofont.icofont-refresh.reload-card
    .card-block
      a.chart
        iframe(src='http://133.186.244.163:50077/#/notebook/2G84Y3CMG/paragraph/20210608-223417_338745986?asIframe', frameborder='0', allowfullscreen)
.col-md-12.col-xl-12
  .card
```

Show graph by using iframe
Make maximize/minimize/refresh button

Code

```
.col-md-12.col-xl-6
  .card.add-task-card
    .card-header
      .card-header-left
        h5 Apart Price in next month
      .card-block
        iframe(src='http://133.186.244.163:50077/#/notebook/2G84Y3CMG/paragraph/20210605-175407_8
// Data widget End
// Required Jquery
script(type='text/javascript' src='/assets/js/jquery/jquery.min.js')
script(type='text/javascript' src='/assets/js/jquery-ui/jquery-ui.min.js')
script(type='text/javascript' src='/assets/js/popper.js/popper.min.js')
script(type='text/javascript' src='/assets/js/bootstrap/js/bootstrap.min.js')
// jquery slimscroll js
script(type='text/javascript' src='/assets/js/jquery-slimscroll/jquery.slimscroll.js')
// modernizr js
script(type='text/javascript' src='/assets/js/modernizr/modernizr.js')
// am chart
script(src='/assets/pages/widget/amchart/amcharts.min.js')
script(src='/assets/pages/widget/amchart/serial.min.js')
// Custom js
script(type='text/javascript' src='/assets/pages/dashboard/custom-dashboard.js')
script(type='text/javascript' src='/assets/js/script.js')
script(type='text/javascript' src='/assets/js/SmoothScroll.js')
script(src='/assets/js/pcoded.min.js')
script(src='/assets/js/demo-12.js')
script(src='/assets/js/jquery.mCustomScrollbar.concat.min.js')
script.
  var $window = $(window);
  var nav = $('.fixed-button');
  $window.scroll(function(){
  if ($window.scrollTop() >= 200) {
  nav.addClass('active');
  }
  else {
  nav.removeClass('active');
  }
  });
```

Show ML result by using iframe

Code

```
0 * * * * /home/hadoop/project_web_advanced/crawling/test/bin/python3
 /home/hadoop/project_web_advanced/crawling/daum_news.py
```

```python
import requests
import urllib.parse as parse
from bs4 import BeautifulSoup
from subprocess import PIPE, Popen
import json

headers = [
    {'User-Agent' : 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) Appl
 (KHTML, like Gecko) Chrome/90.0.4430.85 Safari/537.36'},
]

BASE_URL = "https://realestate.daum.net"
url = "/news/region/"
base_url = BASE_URL + url

region_list = {
    '전체': 'nation',
    '서울': 'seoul',
    '경기': 'gyeonggi',
    '인천': 'incheon',
    '부산': 'busan',
    '대구': 'daegu',
    '광주': 'gwangju',
    '대전': 'daejeon',
    '세종': 'sejong',
    '울산': 'ulsan',
    '강원': 'gangwon',
    '충청': 'chungcheong',
    '경상': 'gyeongsang',
    '전라': 'jeolla',
    '제주': 'jeju'
}
```

```python
for region in region_list.values():
    data = requests.get(base_url+region, params=params, headers=headers[0])
    html = data.content.decode('utf-8')
    soup = BeautifulSoup(html, 'html.parser')
    #mCenter > div.wrap_partnews > ul > li.fst > div > strong > a
    #mCenter > div.wrap_partnews > ul > li:nth-child(3) > div > strong > a
    news_list = soup.select('#mCenter > div.wrap_partnews > ul > li')

    json_data_list = []

    for news in news_list:
        data = {
            'title' : news.div.strong.a.string,
            'url' : BASE_URL + news.div.strong.a.attrs['href']
        }
        json_data_list.append(data)

    json_data = {
        'news' : json_data_list
    }
    json_data_list = json.dumps(json_data)
    file_name = region+'.json'
    f = open('/home/hadoop/project_web_advanced/crawling/data/'+file_name,
    f.write(json_data_list)
    f.close()
```

Save news data in json format

Execute code every hour using crontab

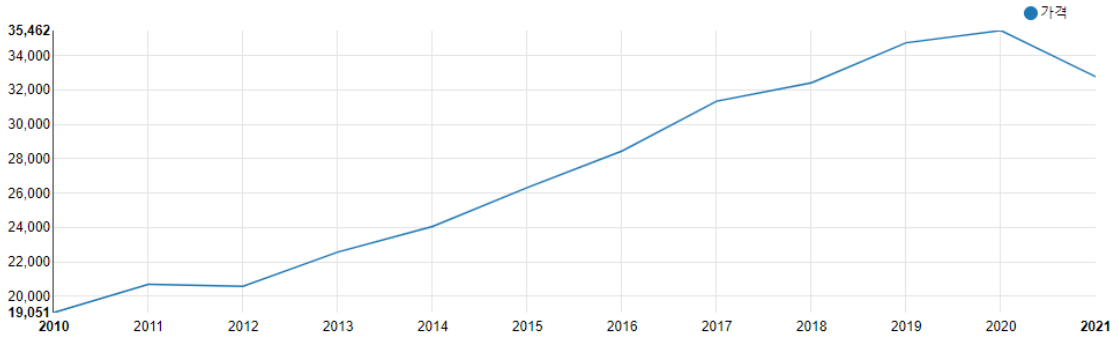
Code

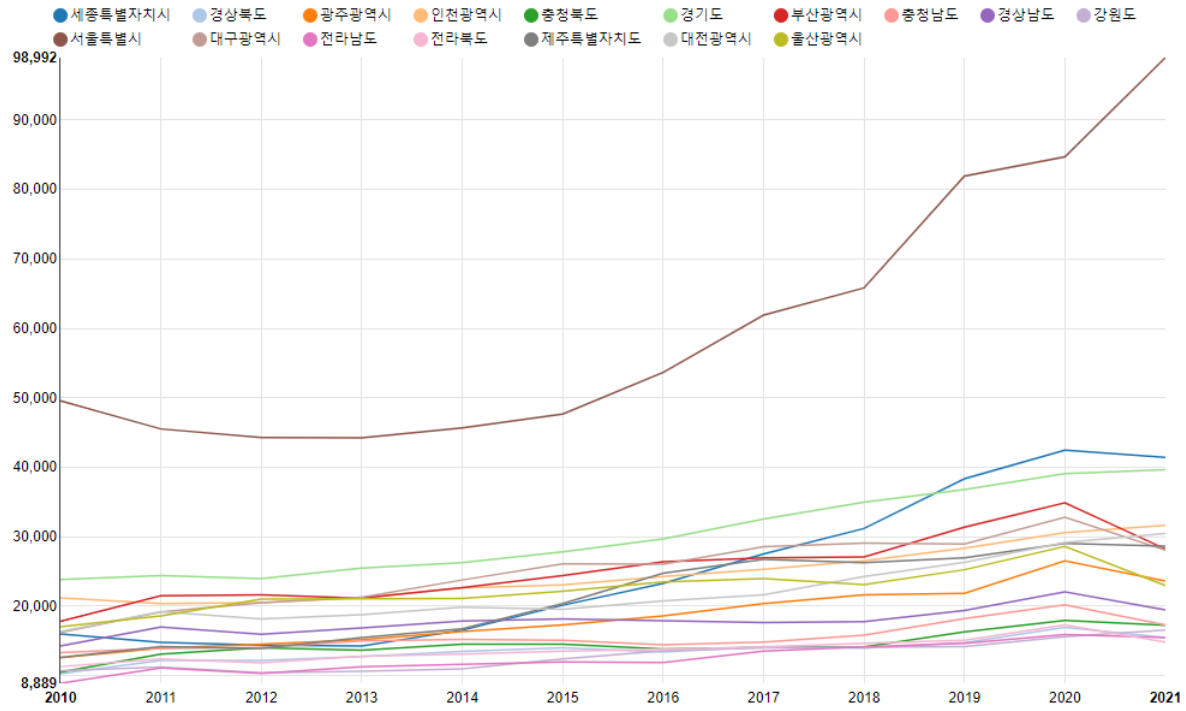
```
var express = require('express');
const fs = require('fs');
var router = express.Router();

/* GET home page. */
router.all('/', function(req, res, next) {
  const path = '/home/hadoop/project_web_advanced/crawling/data/';

  var nationData = fs.readFileSync(path+'nation.json', 'utf8');
  nationData = JSON.parse(nationData);

  var seoulData = fs.readFileSync(path+'seoul.json', 'utf8');
  seoulData = JSON.parse(seoulData);

  var gyeonggiData = fs.readFileSync(path+'gyeonggi.json', 'utf8');
  gyeonggiData = JSON.parse(gyeonggiData);

  var incheonData = fs.readFileSync(path+'incheon.json', 'utf8');
  incheonData = JSON.parse(incheonData);

  var busanData = fs.readFileSync(path+'busan.json', 'utf8');
  busanData = JSON.parse(busanData);

  var daeguData = fs.readFileSync(path+'daegu.json', 'utf8');
  daeguData = JSON.parse(daeguData);

  var gwangjuData = fs.readFileSync(path+'gwangju.json', 'utf8');
  gwangjuData = JSON.parse(gwangjuData);

  var daejeonData = fs.readFileSync(path+'daejeon.json', 'utf8');
  daejeonData = JSON.parse(daejeonData);
```

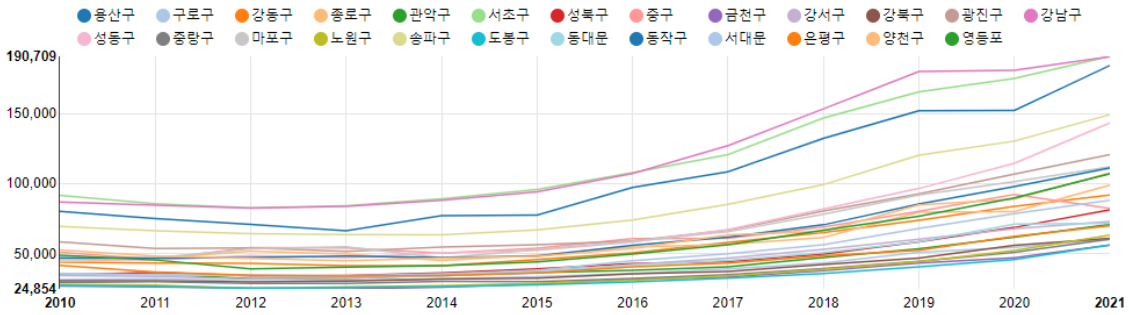
```
.card-block-tab
  .tabmenu
    ul
      li#tab1.btnCon
        input#tabmenu1(type='radio' checked='' name='tabmenu')
        label(for='tabmenu1') 전체
        .tabCon
          - const nation = nation_news;
            each n in nation
              a(href=n.url) #{n.title}
              br
      li#tab2.btnCon
        input#tabmenu2(type='radio' checked='' name='tabmenu')
        label(for='tabmenu2') 서울
        .tabCon
          - const seoul = seoul_news;
            each n in seoul
              a(href=n.url) #{n.title}
              br
      li#tab3.btnCon
        input#tabmenu3(type='radio' checked='' name='tabmenu')
        label(for='tabmenu3') 경기
        .tabCon
          - const gyeonggi = gyeonggi_news;
            each n in gyeonggi
              a(href=n.url) #{n.title}
              br
```

Load news json file in routes/index.js
Show news by region on the web

Result

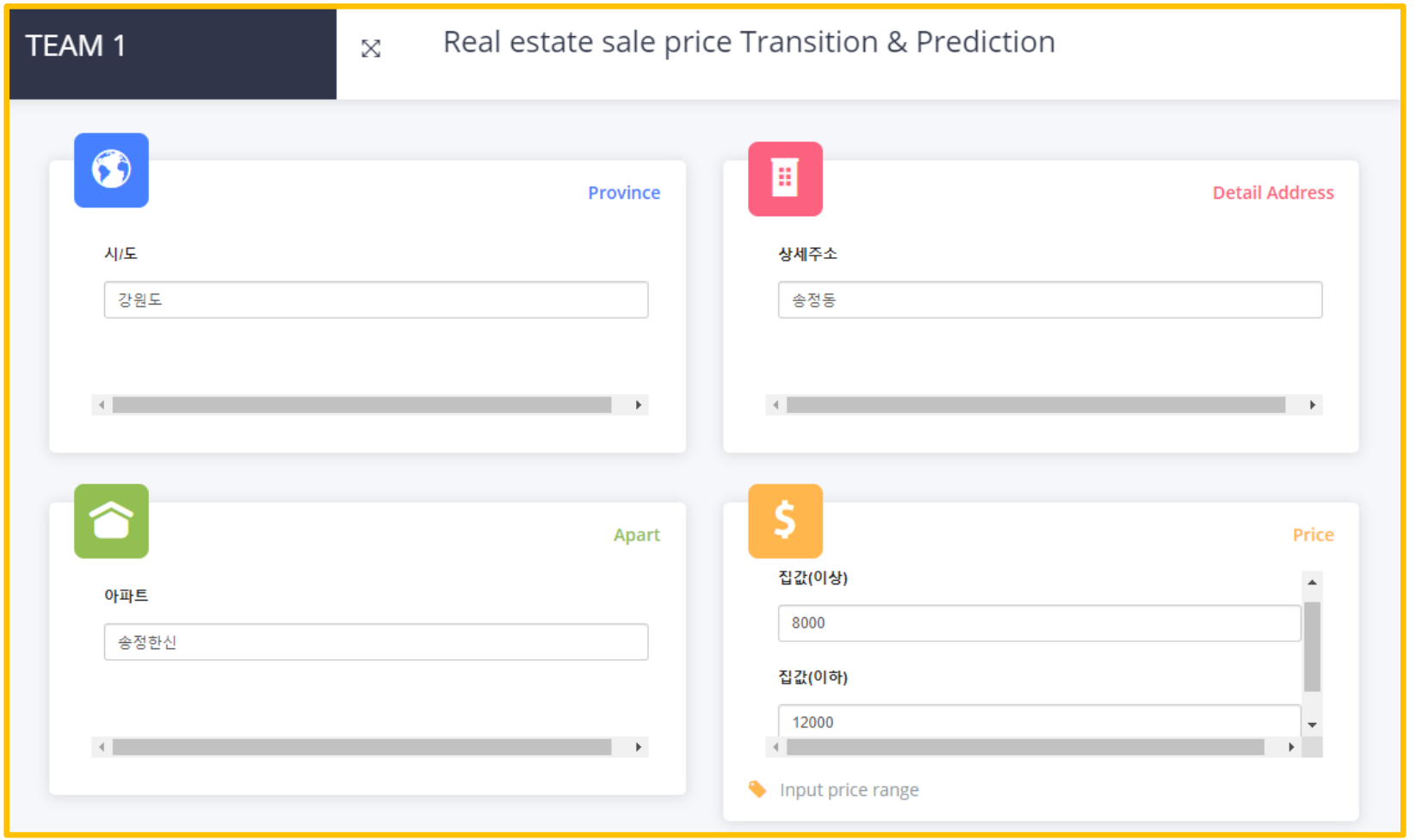






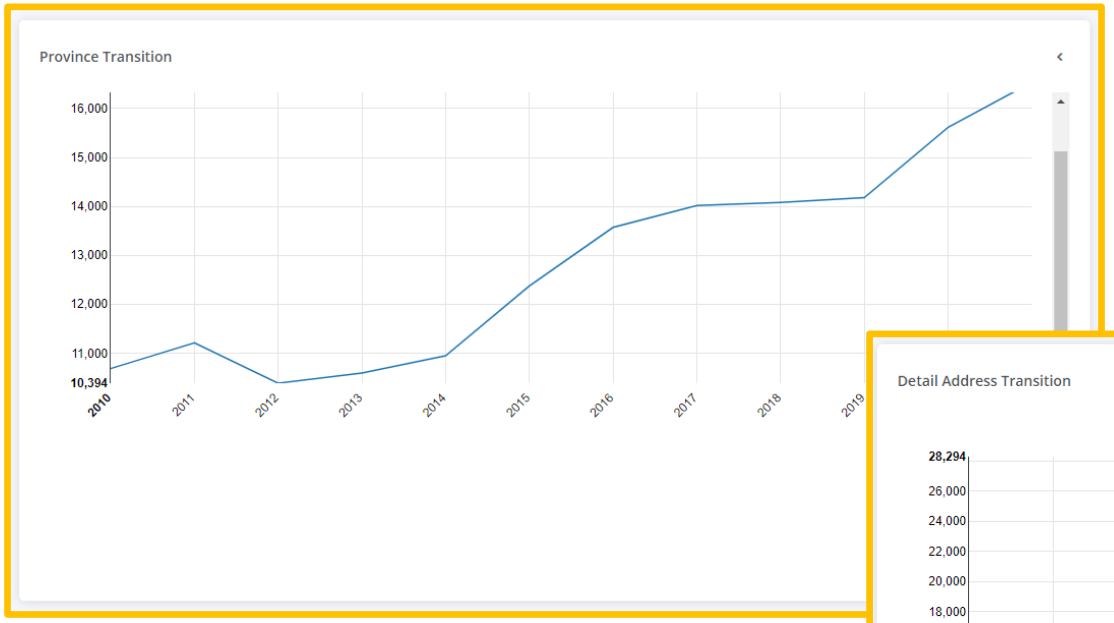
Overall trend

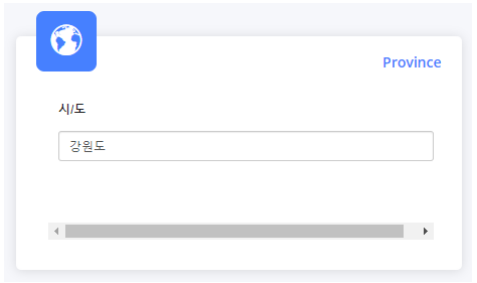
Result

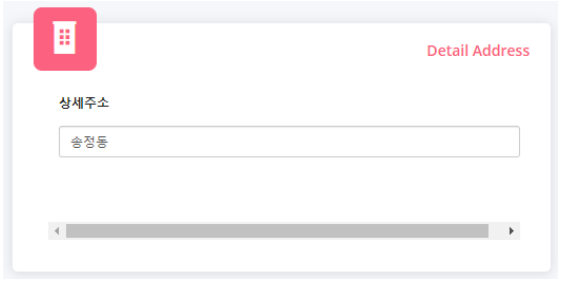


Input “province”, “detail address”, “apart name”, “price range”

Result

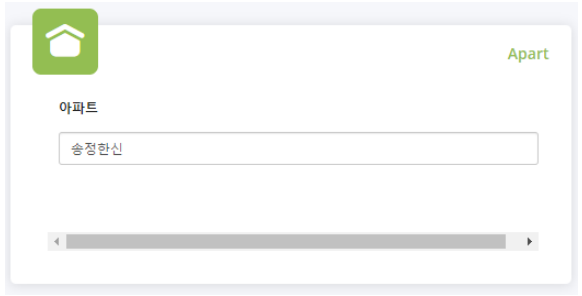


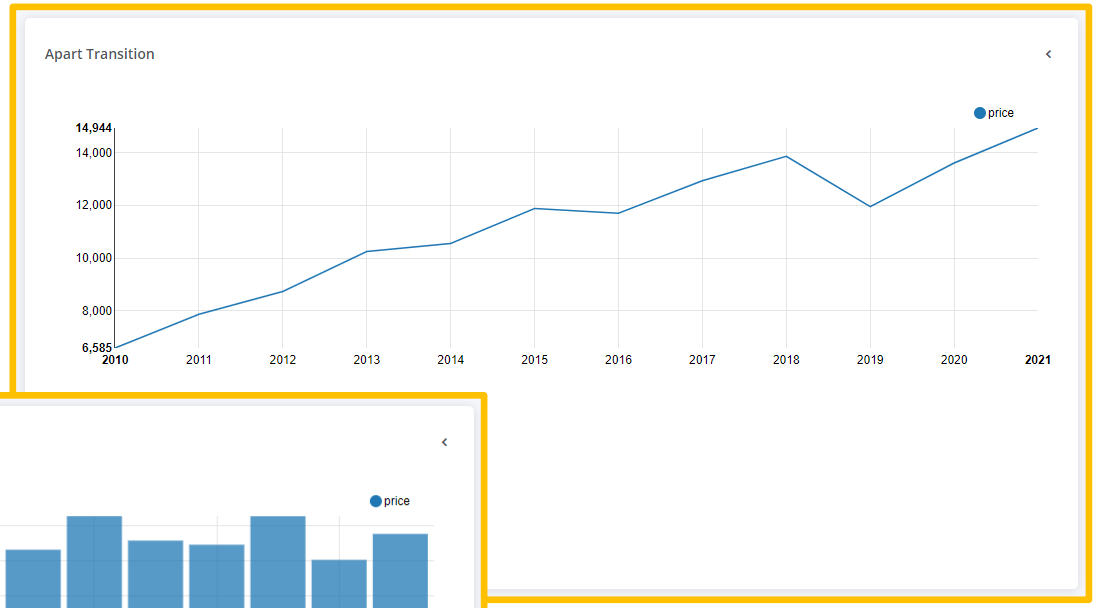


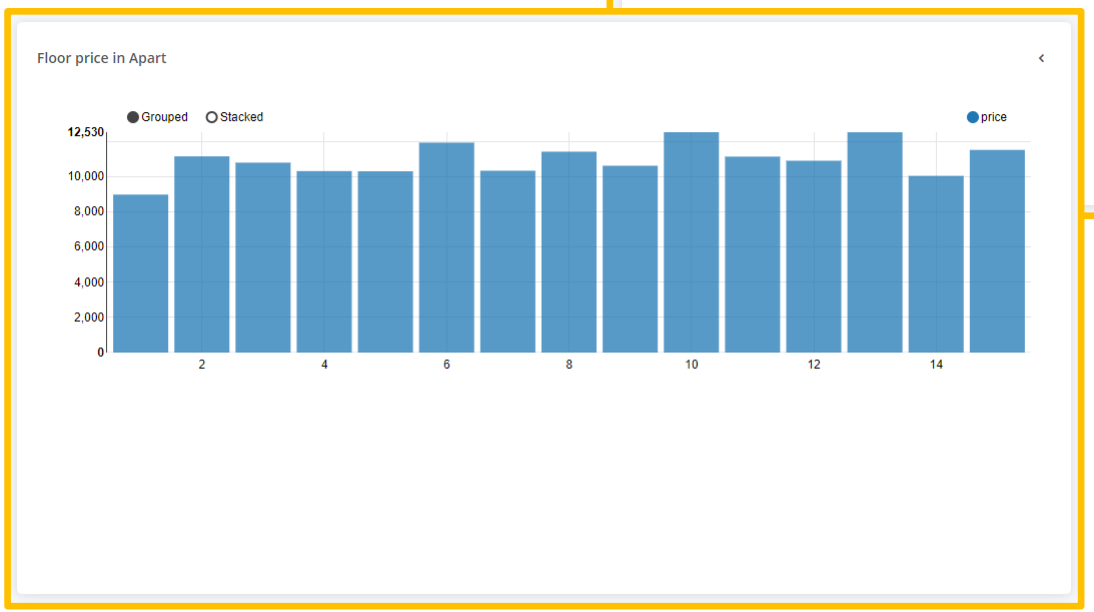


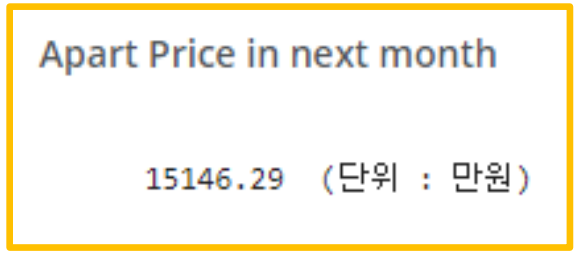
Graph about housing price change by “province” and “detail address”

Result



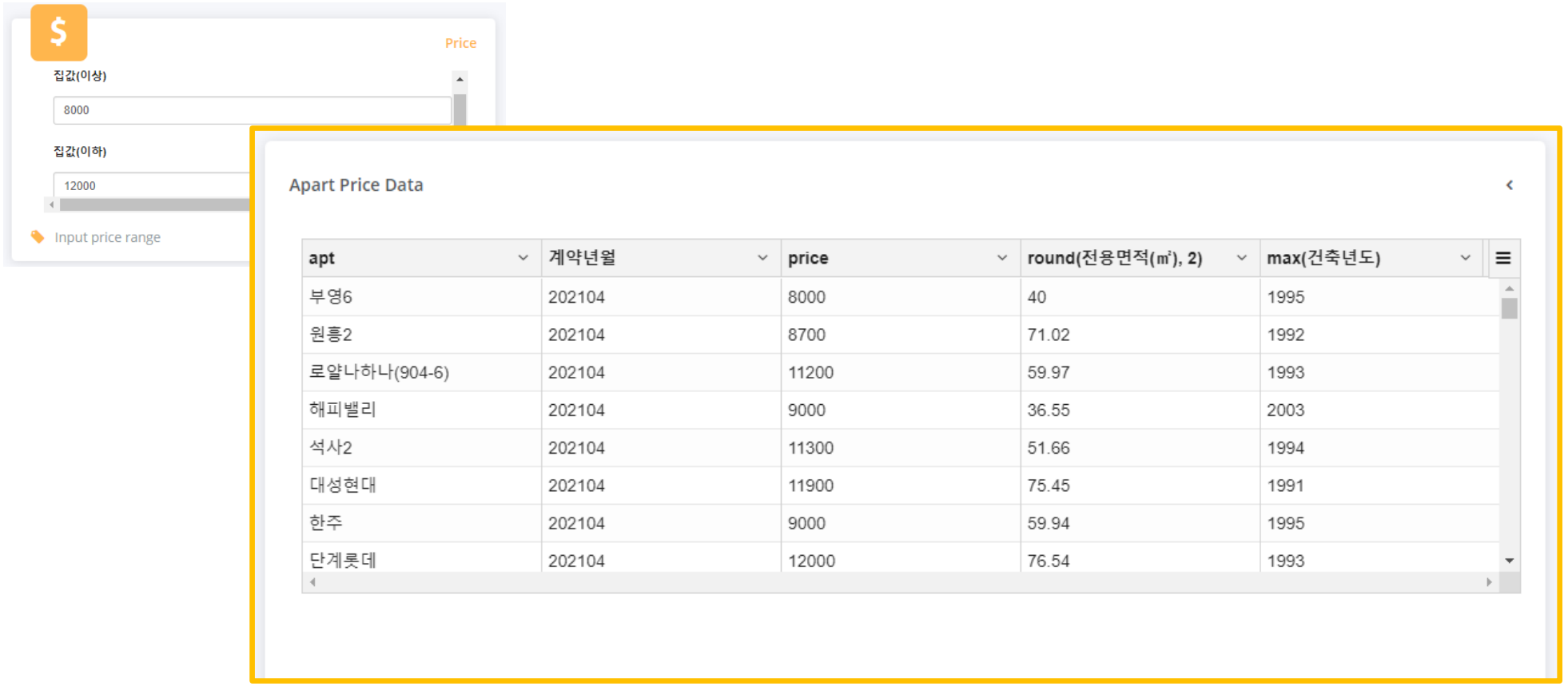






Graph about housing price change by “apart name” and “floor”

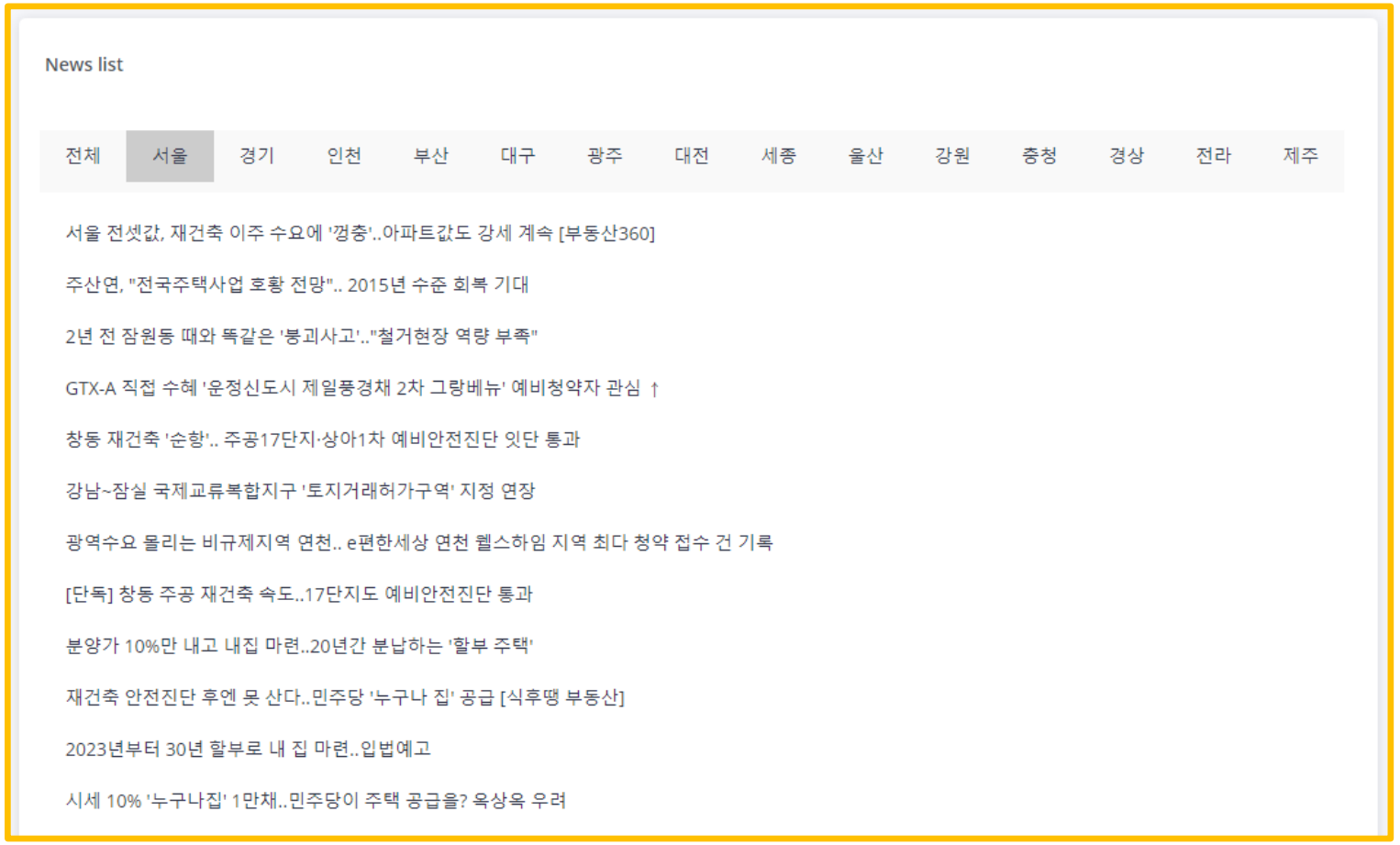
Result



apt	계약년월	price	round(전용면적(㎡), 2)	max(건축년도)
부영6	202104	8000	40	1995
원흥2	202104	8700	71.02	1992
로얄나하나(904-6)	202104	11200	59.97	1993
해피밸리	202104	9000	36.55	2003
석사2	202104	11300	51.66	1994
대성현대	202104	11900	75.45	1991
한주	202104	9000	59.94	1995
단계롯데	202104	12000	76.54	1993

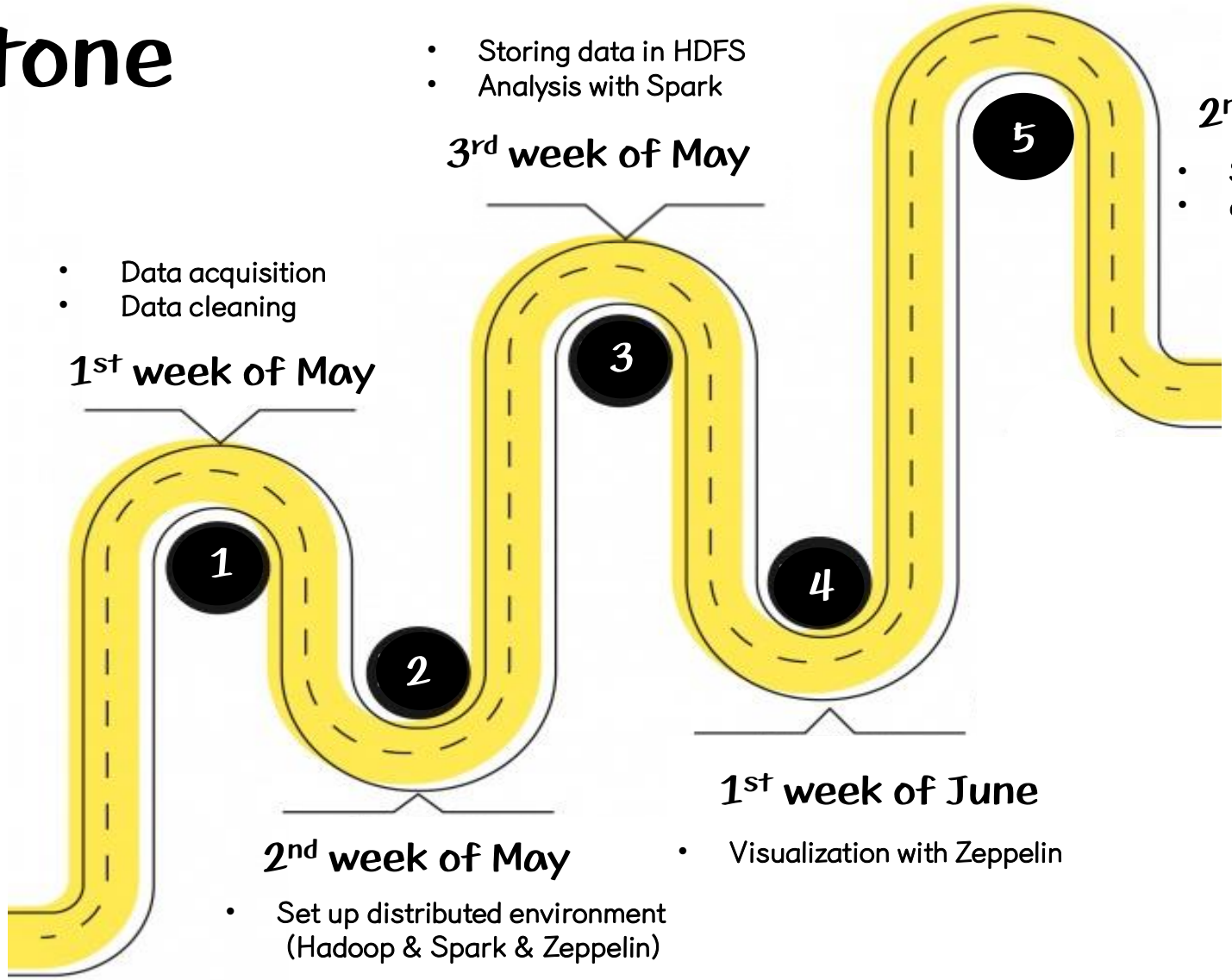
Search record by “price range”

Result



News tab for showing news by region

Milestone



1st week of May

- Data acquisition
- Data cleaning

2nd week of May

- Set up distributed environment (Hadoop & Spark & Zeppelin)

3rd week of May

- Storing data in HDFS
- Analysis with Spark

1st week of June

- Visualization with Zeppelin

2nd week of June

- Set up website and
- display final results on it

Thank You!