서울시 대기오염 측정정보 분석 (2013-2022)

Seaborn, Matplotlib, Pandas, PySpark

Contents

- Exploratory Data Analysis
- Data Preprocessing
- Analysis using Seaborn, Matplotlib, Pandas, PySpark
- Reflecting
- Appendix

Exploratory Data Analysis(서울 열린데이터 광장)

구현 환경

공공데이터 10년치로 **행의 수가 92,028,293개**인 데이터를 다루었습니다.

AWS EC2 3개의 instance를 **clustering**하여 **Pyenv 환경**에서 **Jupyter lab을 사용하여 Spark**로 구현하였습니다.

그리고 시각화를 위해 **DataFrame, Seaborn, Matplotlib, Pandas**를 사용하였습니다.

Exploratory Data Analysis(서울 열린데이터 광장)

항목	정보
측정일시	측정한 일시 (long type, e.g. 2022010100)
측정소 코드	각 자치구의 측정소에 해당하는 코드 (long type)
측정항목	대기오염 측정정보의 항목 구분값 (long type)
평균값	1시간 평균 측정값을 보정한 값 (long type)
측정기 상태	“0”: 정상, 그 외 : 기타 (long type)
국가 기준초과 구분	“0”: 정상, “1” : 초과 (long type)
지자체 기준초과 구분	“0”: 정상, “1” : 초과 (long type)

Exploratory Data Analysis(서울 열린데이터 광장)

가설

1. 환경에 대한 이슈가 늘 대두되는 상황이기 때문에 대기오염에 대한 10년간 축적 데이터에서 점차 오염도가 증가하는 추세를 보일 것이다.
2. 계절의 영향을 받아 온도가 높은 여름에 오염도가 높을 것이다.

⇨ **통계적 자료를 시각화하여 확인**

Data Preprocessing

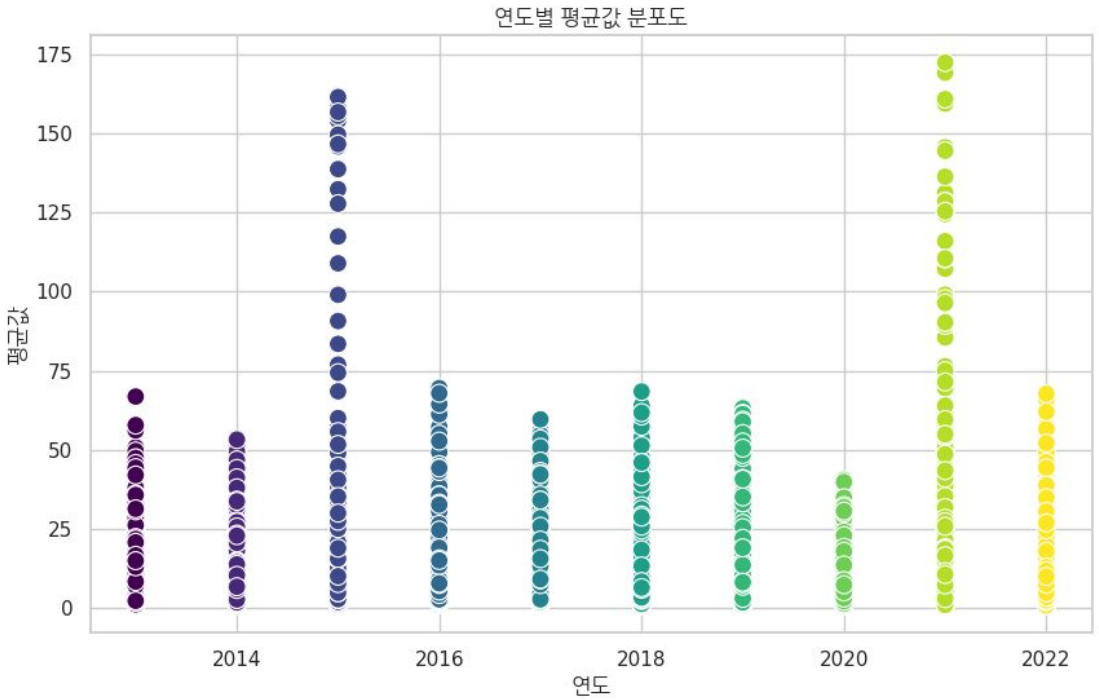
```
import pyspark
from pyspark.sql import SparkSession

spark = SparkSession.builder.appName("AirPollution").getOrCreate()
dataframe =
spark.read.parquet("/home/ubuntu/work/spark_pollution/AirPollution.parquet")
```

```
from pyspark.sql.functions import col, avg, substring, cast

# 측정기 상태가 0인 경우 필터링
filtered_df = dataframe.filter(col("측정기 상태") == 0)

# 중복된 측정일시 기준으로 평균값 계산
grouped_df = filtered_df.groupBy("측정일시").agg(avg("평균값").alias("평균값"))

# 연도 및 월 추출
grouped_df = grouped_df.withColumn("연도", (substring(col("측정일시").cast("string"), 1,
4)).cast("int"))
grouped_df = grouped_df.withColumn("월", (substring(col("측정일시").cast("string"), 5, 2)).cast("int"))

# 월별 평균값 계산
monthly_avg = grouped_df.groupBy("연도", "월").agg(avg("평균값").alias("평균값"))
```

Read DataSet

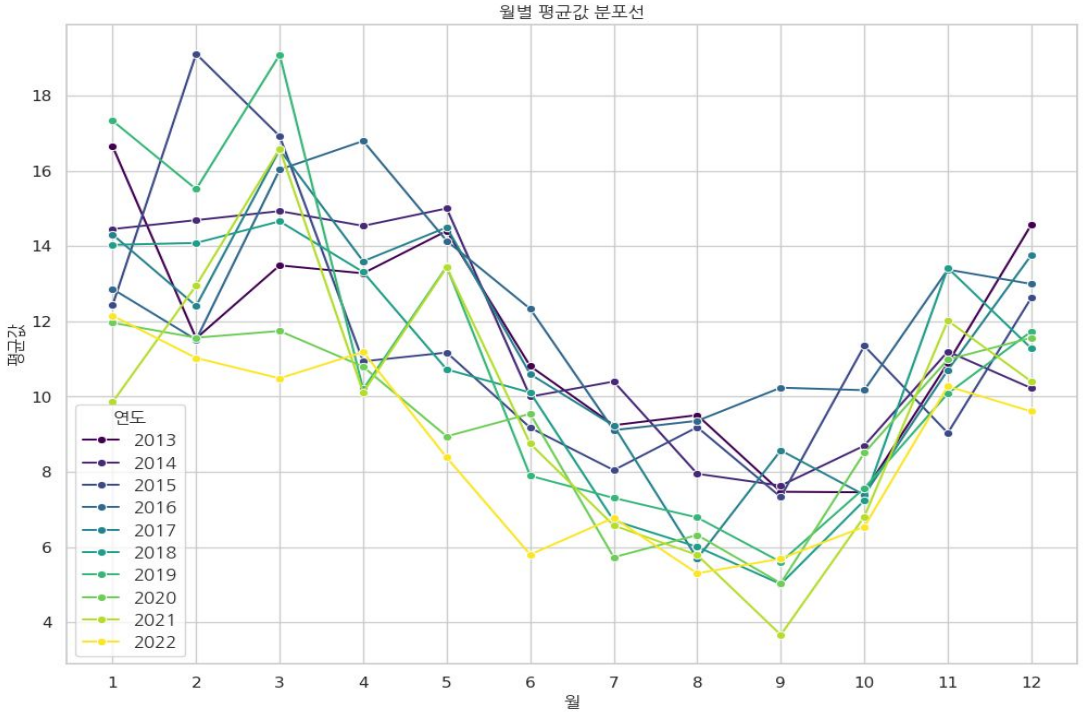
Extract Data

Analysis using Seaborn, Matplotlib, Pandas, PySpark



가설과 달리, 서울시 대기오염 측정정보에 따르면 **연도에 따라** 점차 **오염도가 증가하지는 않았음**을 파악할 수 있었습니다.

Analysis using Seaborn, Matplotlib, Pandas, PySpark



계절에 따른 대기오염의 영향을 파악할 수 있었습니다.

여름에 대기오염도가 **낮아**지며, **겨울**에 대기오염도가 **높아**집니다.

Reflecting

Seaborn, Matplotlib, Pandas, PySpark을 통해 **Spark**를 공부한 내용을 남기고 싶었습니다. **시각화 도구**를 사용하였더니 **시간, 방향 효율적인 데이터 분석**을 할 수 있었습니다.

가설에 따른 데이터 분석 방향성을 잡았지만 가설이 맞지 않을 수 있기 때문에 일부 가공한 데이터 결과를 시각화 해보았습니다. 시각화(Seaborn)로 인해 가설과 다른 결과라는 이슈 발생가 발생했습니다.

가설과 다른 결과를 반영하여 방향성을 재구성하였습니다.

Appendix

Code(github)