
서울시 미세먼지 정책
:수소버스 효율적 운용방안

활용데이터정의

Data Definition

활용 데이터 출처 및 정의

데이터 종류	데이터 출처
서울시 기상데이터 (일별, 지역구별)	기상청, https://bd.kma.go.kr/kma2017/
서울시 대기오염 데이터	서울특별시 대기환경정보, http://cleanair.seoul.go.kr
서울시 유동인구 데이터	서울시 열린데이터광장(유동인구, 정류소ID, 좌표)

(2017년 데이터 기준)

Tools	Packages
R(V3.5.1) Python	wordcloud(데이터 마이닝) ggplot2(데이터 시각화) leaflet(데이터 지도화) dplyr(전처리, 데이터 핸들링) corrplot(상관관계) dygraphs(시계열 그래프) ROSE(오버 샘플링) ROCR(ROC curve, auc) Stringr(전처리) Arules H20(RF, DNN 모델링)

활용 변수 소개

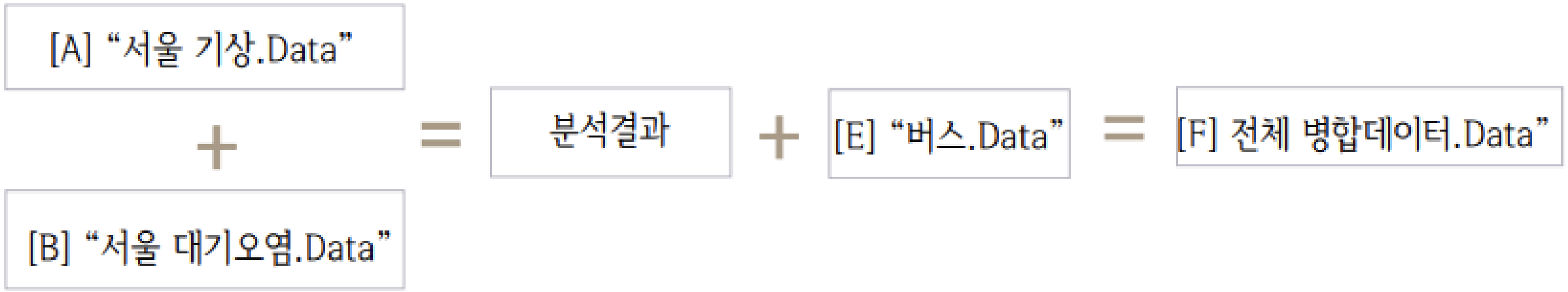
변수명	형식
day	datetime
Temp(온도)	numeric
rainfall(강수량)	numeric
W_s(풍속)	numeric
humi(습도)	numeric
dew(이슬점 온도)	numeric
press(압력)	numeric
rand_temp(지면압력)	numeric
See_1(시정)	category
W_d(풍향)	factor

변수명	형식
이산화질소	numeric
오존 농도	numeric
일산화탄소	numeric
아황산가스 농도	numeric
미세먼지 농도	numeric
초미세먼지 농도	numeric

데이터처리방안

Data Preprocessing

탐색적 자료분석(EDA) - 데이터 병합과정

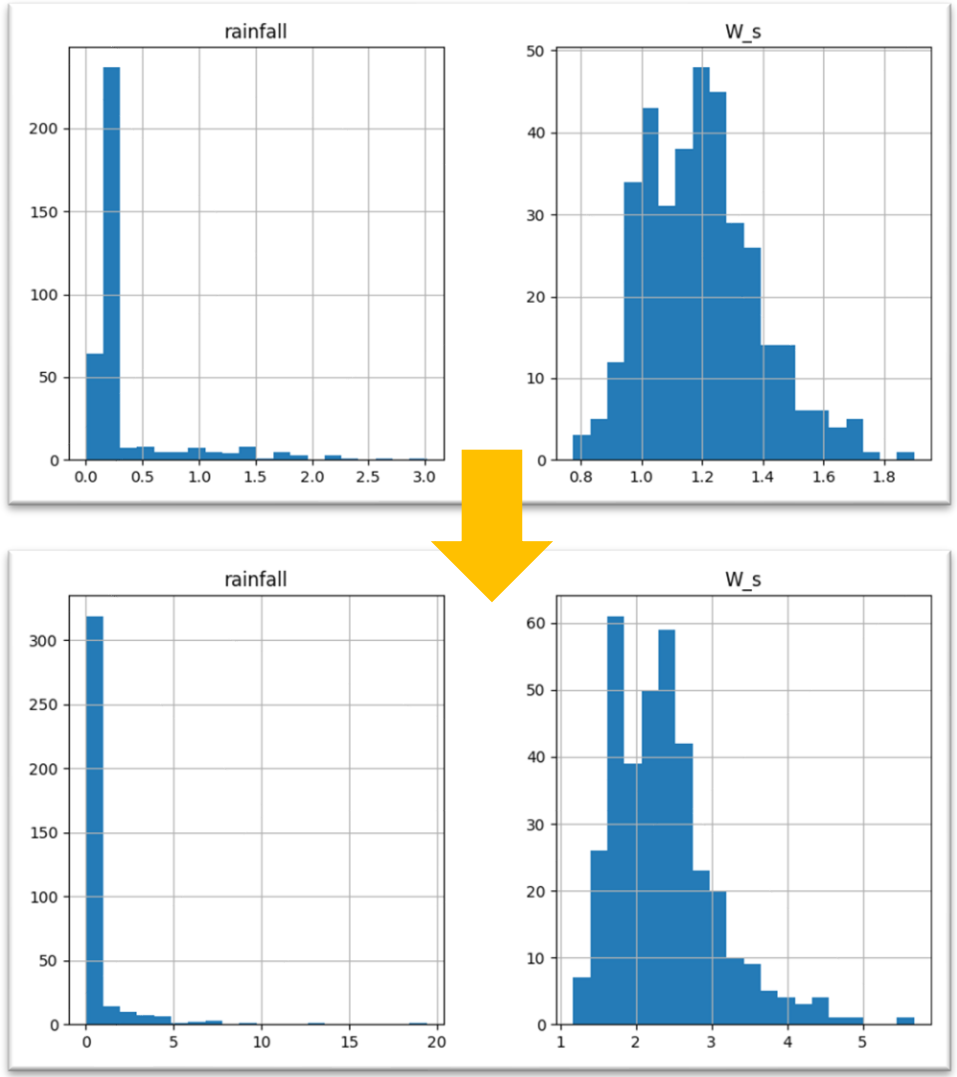


데이터 처리방안 - PCA & 다중공성제거

2022 서울시 기상데이터 (기상청)

- 일 기준 그룹화
  - 수치형 : mean
  - 명목형 : mode

- 로그 변환 : 정규성을 만족시켜주기 위해
  - rainfall
  - W_s



데이터 처리방안 - PCA & 다중공성제거

활동 데이터

: 2022 서울시 대기오염 데이터 (출처:서울 열린데이터 광장)

전처리 과정

(1) 연속형 변수 결측치 존재 → 중위값 대체

(2) '측정소명' 열 삭제

(3) group.by('측정일시').mean()

(4) '측정일시' datetime type으로 변경

	측정일시	이산화질소농도(ppm)	오존농도(ppm)	일산화탄소농도(ppm)	아황산가스농도(ppm)	미세먼지농도(㎍/㎥)	초미세먼지농도(㎍/㎥)
0	20220101	0.03412	0.01278	0.624	0.00352	27.80	14.38
1	20220102	0.02966	0.01638	0.612	0.00334	35.32	23.58
2	20220103	0.03846	0.00894	0.660	0.00360	25.90	15.04
3	20220104	0.02618	0.01678	0.594	0.00372	35.00	19.60
4	20220105	0.04180	0.00664	0.812	0.00404	48.26	28.82
...	...	...	...	...	...	...	...
360	20221227	0.04670	0.00826	0.872	0.00366	49.08	36.00
361	20221228	0.02792	0.01738	0.564	0.00352	34.92	20.46
362	20221229	0.03286	0.01290	0.576	0.00380	30.10	17.84
363	20221230	0.03366	0.01472	0.606	0.00360	38.20	26.94
364	20221231	0.04198	0.00830	0.786	0.00378	43.18	32.22

365 rows × 7 columns

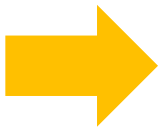
데이터 처리방안 - PCA & 다중공성제거

2022 버스노선별_정류장별_시간대별_승하차 인원정보(서울시 열린데이터 광장)

- **버스정류장별 유동인구**
  월별로 승하차 승객수의 총 합

- **정류장의 밀집도**
  역명을 기준으로 버스노선 수의 총 합

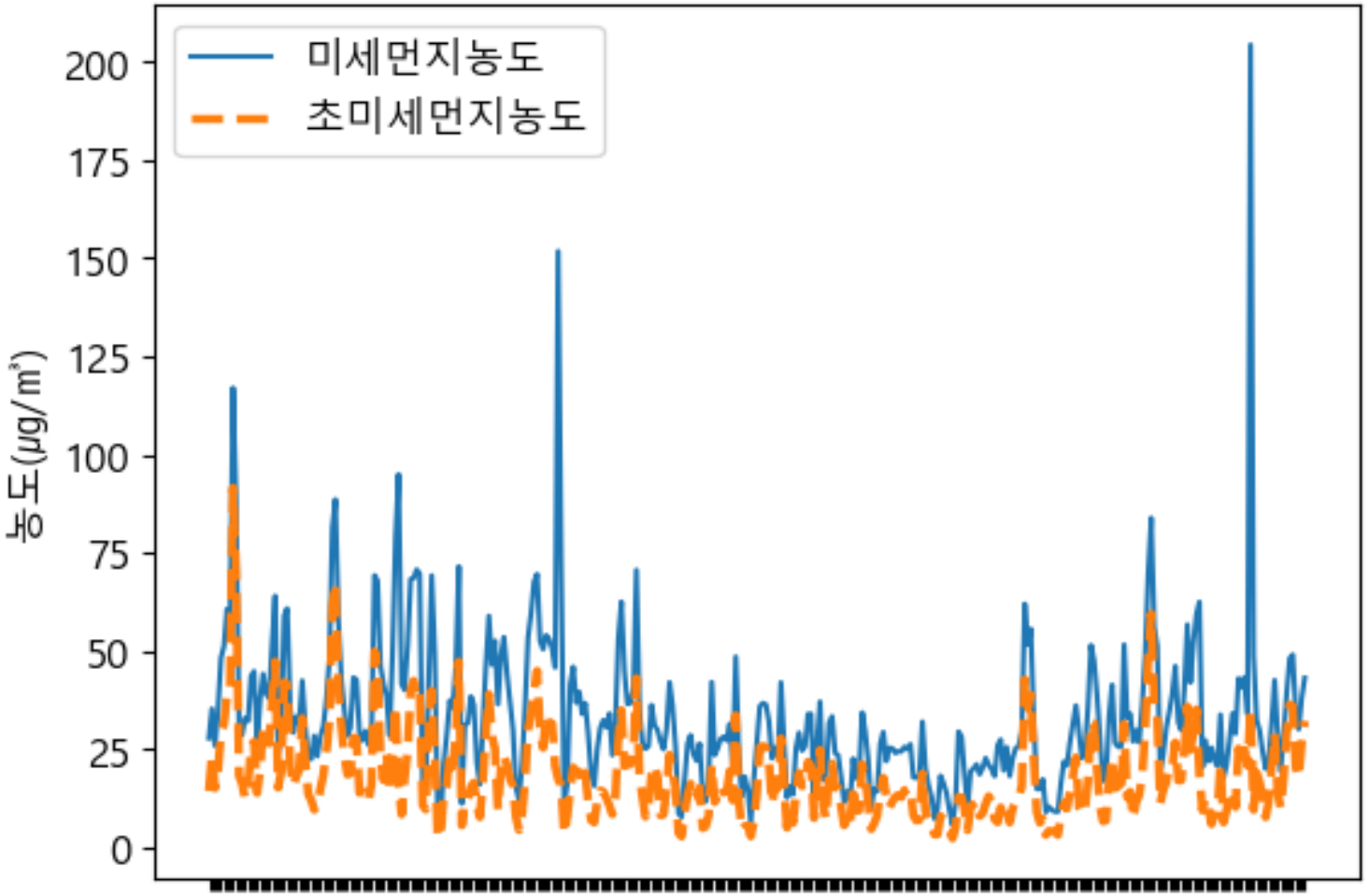
사용년월	노선번호	노선명	표준버스정	버스정류장	역명	00시승차총	00시하차총	1시승차총	1시하차총	2시승차총	2시하차총
202201	741	741번(진관	1E+08	1001	종로2가사거리	27	69	14	40	0	0
202201	N37	N37번(진관	1E+08	1001	종로2가사거리	0	0	0	0	9	8
202201	470	470번(상암	1E+08	1001	종로2가사거리	10	49	0	0	0	0
202201	N37	N37번(송파	1E+08	1001	종로2가사거리	0	5	31	89	12	49
202201	100	100번(하계	1E+08	1002	창경궁.서울대학	0	0	0	0	0	0
202201	107	107번(민락	1E+08	1002	창경궁.서울대학	2	0	0	0	0	0
202201	104	104번(강북	1E+08	1002	창경궁.서울대학	0	0	0	0	0	0
202201	171	171번(용마	1E+08	1002	창경궁.서울대학	0	0	0	0	0	0
202201	172	172번(하계	1E+08	1002	창경궁.서울대학	0	0	0	0	0	0
202201	140	140번(도봉	1E+08	1002	창경궁.서울대학	0	0	0	0	0	0
202201	272	272번(면목	1E+08	1002	창경궁.서울대학	0	0	0	0	0	0
202201	301	301번(장지	1E+08	1002	창경궁.서울대학	0	0	0	0	0	0
202201	102	102번(상계	1E+08	1002	창경궁.서울대학	0	0	0	0	0	0
202201	150	150번(도봉	1E+08	1002	창경궁.서울대학	0	0	0	0	0	0
202201	710	710번(상암	1E+08	1002	창경궁.서울대학	7	9	0	0	0	0
202201	151	151번(우이	1E+08	1002	창경궁.서울대학	0	0	0	0	0	0
202201	106	106번(의정	1E+08	1002	창경궁.서울대학	1	3	0	0	0	0
202201	143	143번(정릉	1E+08	1002	창경궁.서울대학	0	0	0	0	0	0



사용년월	유동인구 수
202201	230284092
202202	199176884
202203	230762464
202204	253268008
202205	273752456
202206	260715325
202207	271184016

지하철_명	노선 수
현대아파트	61
서울역버스환승	60
연희104고지앞	56
숭례문	55
신도림역	55
노량진역	55
동묘앞	54
종로2가	52
논현역	52

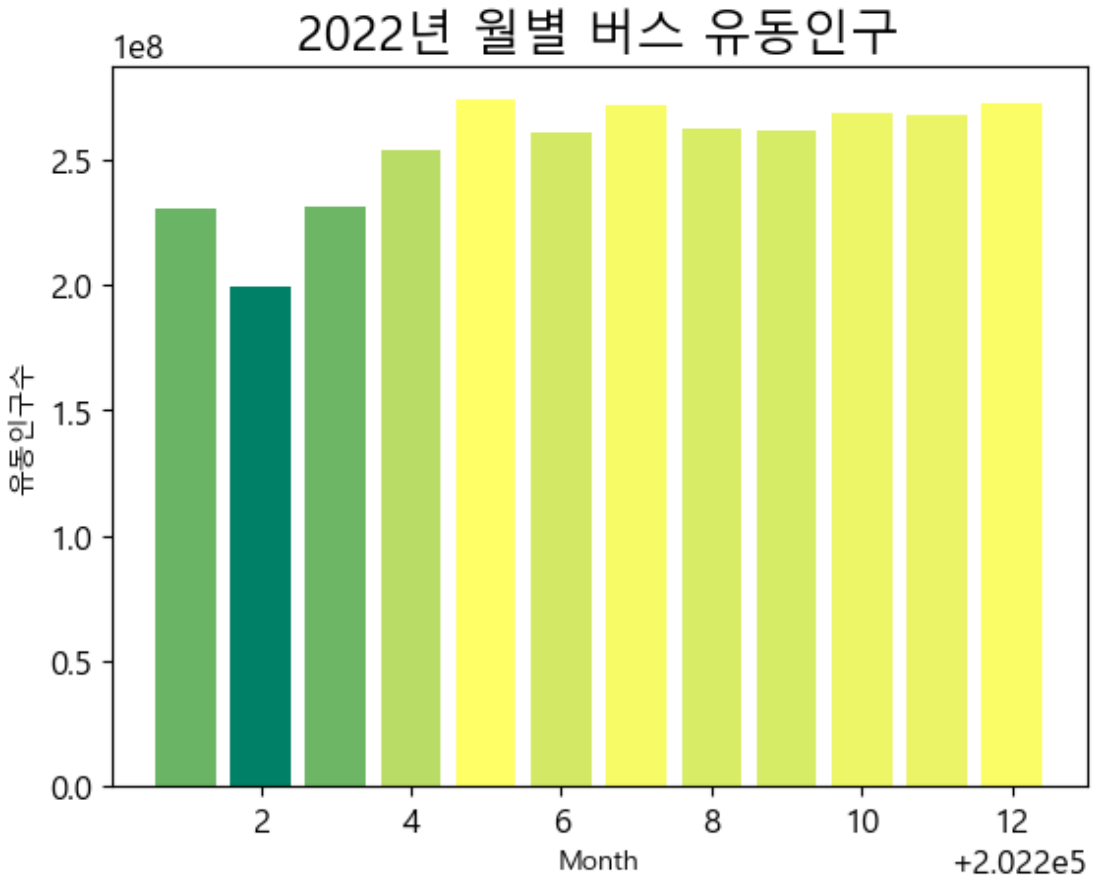
데이터 처리방안 - 미세먼지 데이터 시계열분석

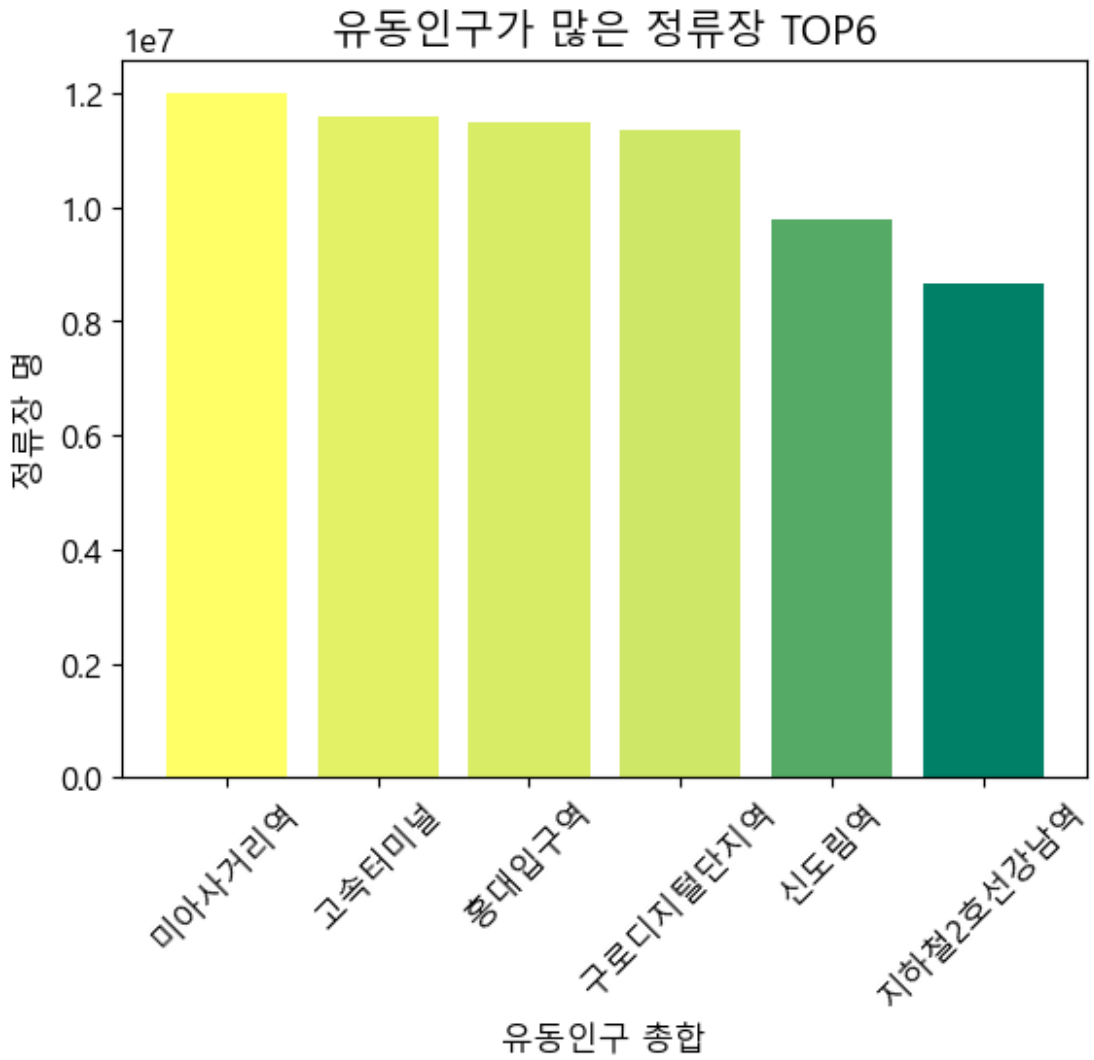


- Target 변수의 생성을 위한 2022년 1월 1일부터 12월 31일까지 발생한 미세먼지(PM10,PM2.5) 시계열 분석.

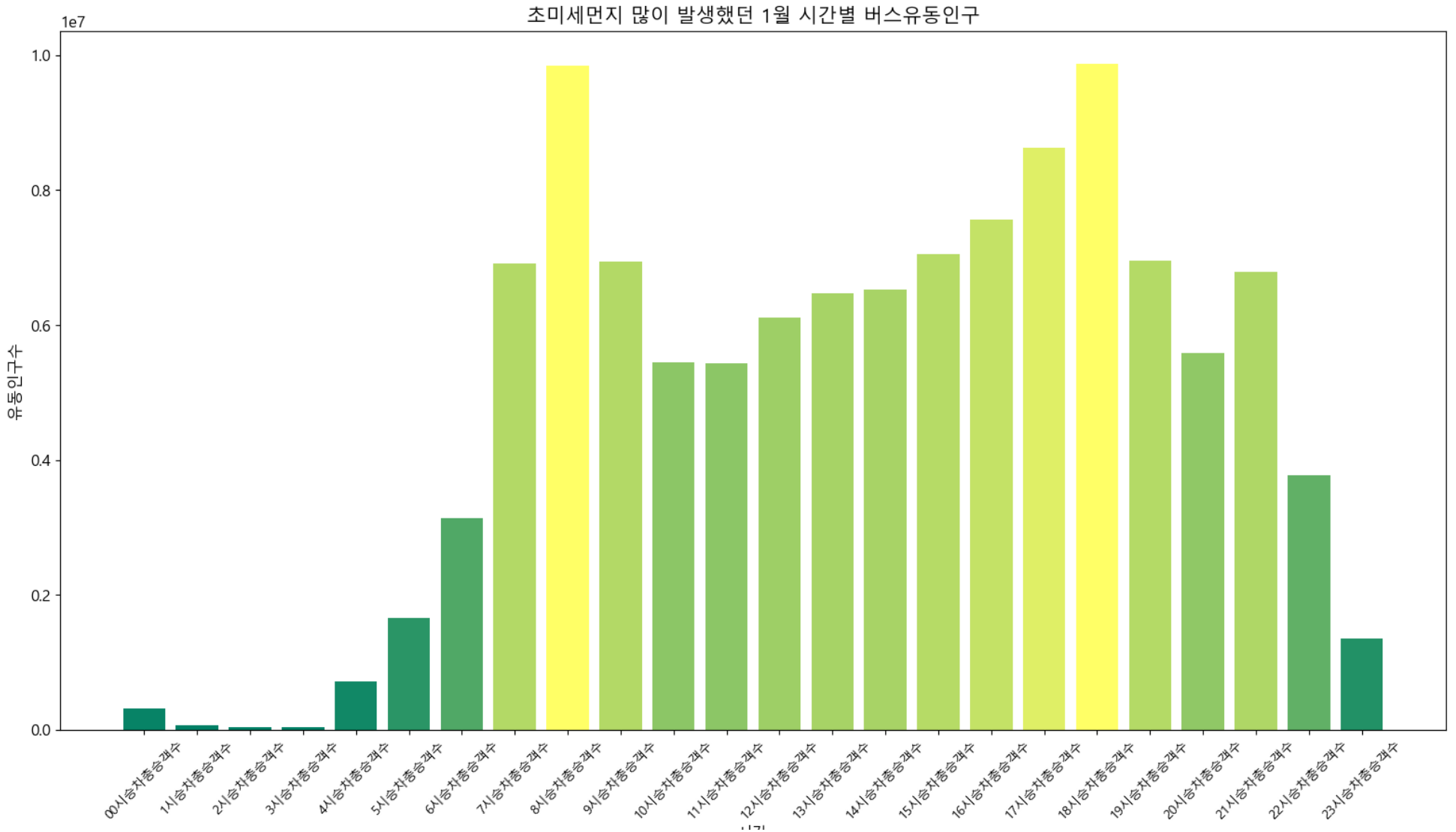
- Target 변수를 PM2.5 35 기준으로 0 또는 1 즉, 미세먼지가 좋다 혹은 나쁘다는 기준을 설정

데이터 처리방안 - 유동인구분석

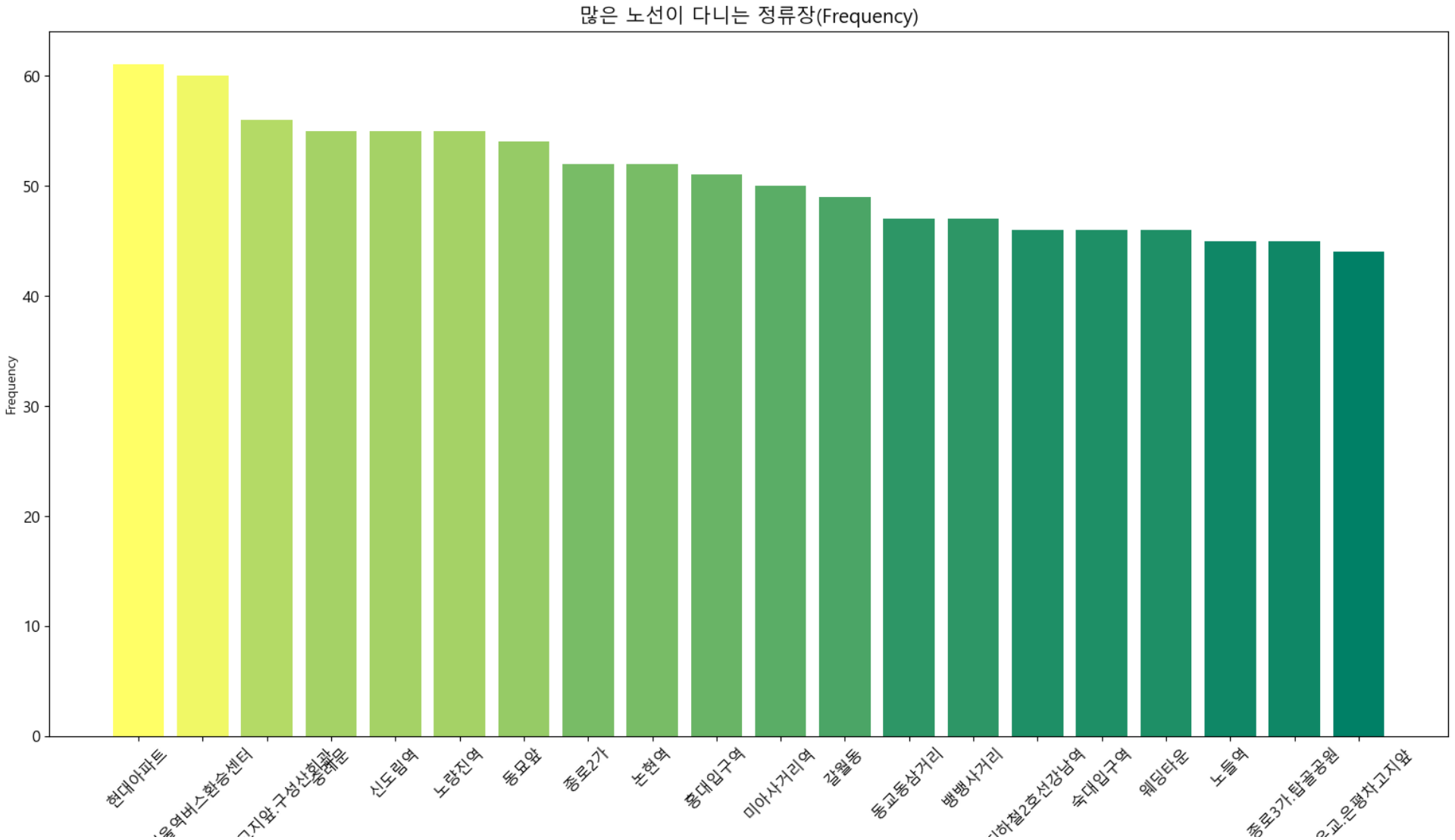




데이터 처리방안 - 유동인구분석



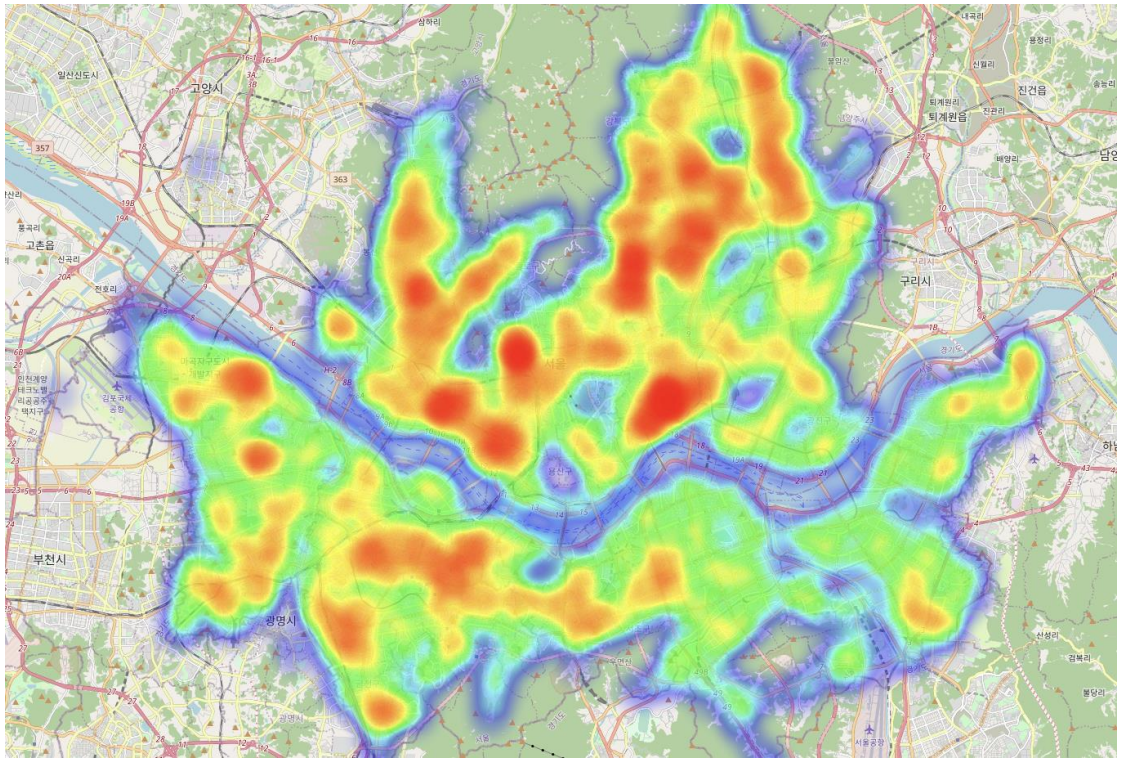
데이터 처리방안 - 버스정류장별 유동인구 분석



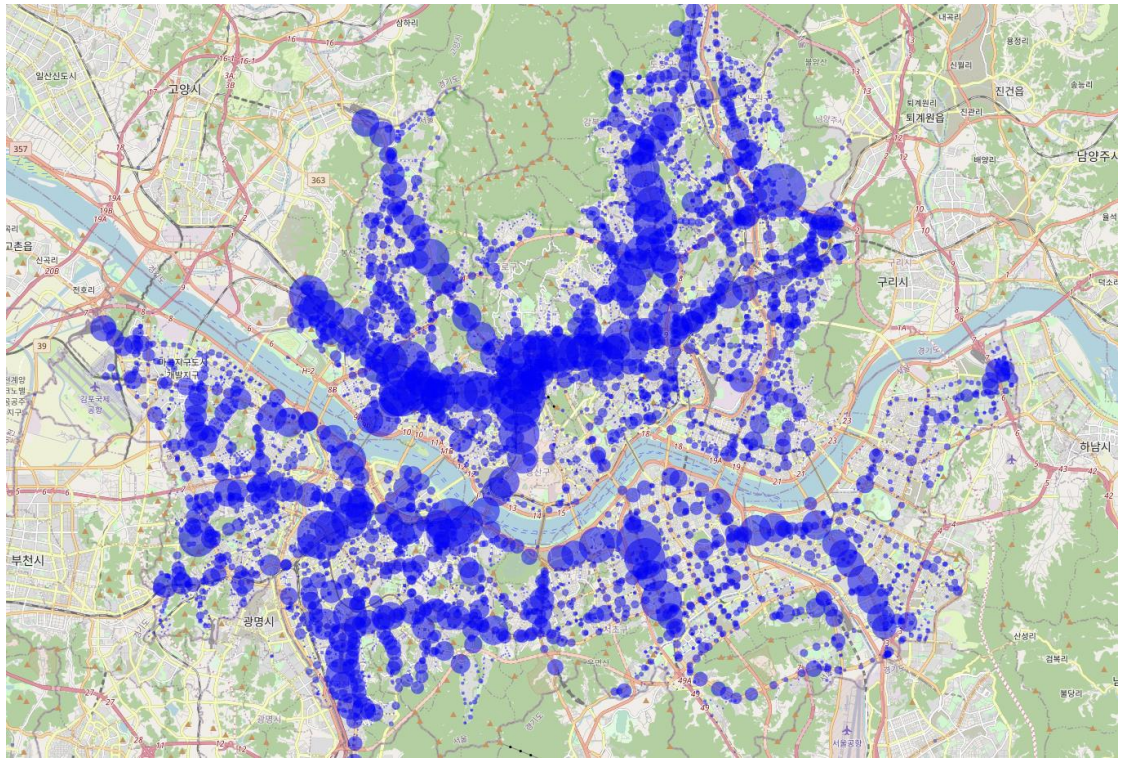
분석 기법

Data modeling

시각화 - 버스 정류소 노선수와 유동인구와의 상관관계 분석



구별 유동인구 밀집도

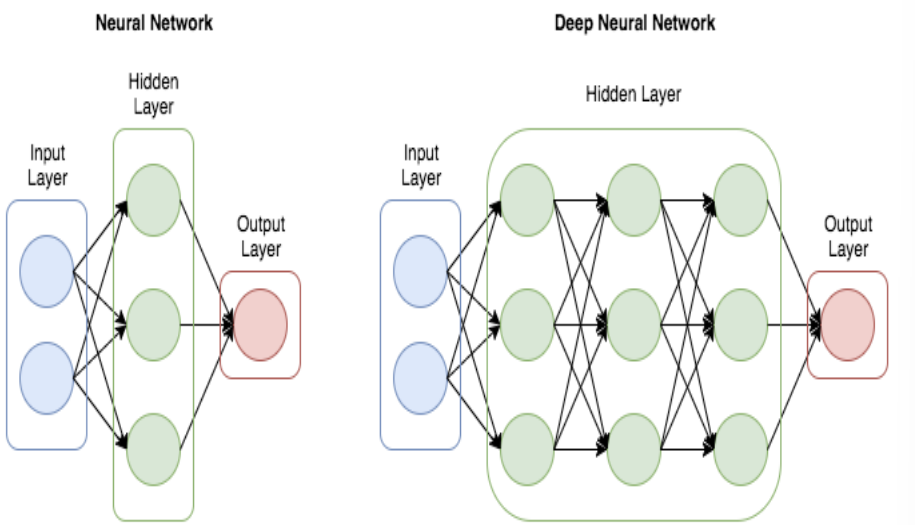


버스정류장밀집도 등고선 시각화

모델링

DNN (Deep Neural Network, 심층신경망)

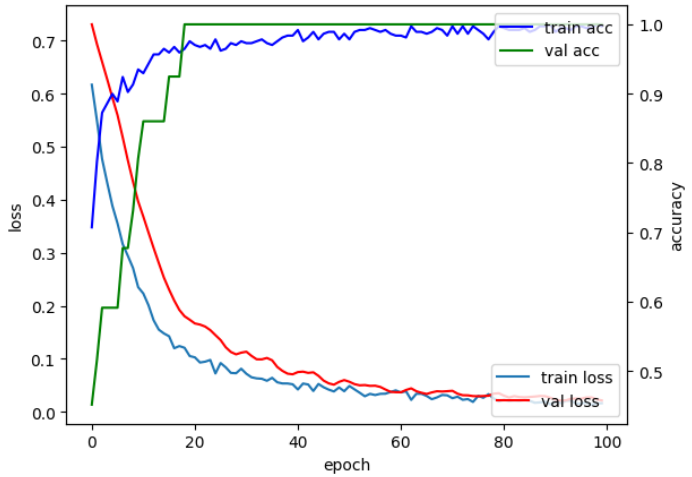
심층 신경망은 입력층과 출력층 사이에 여러 개의 은닉층들로 이루어진 인공신경망으로, 복잡한 비선형 관계들을 모델링할 수 있다.



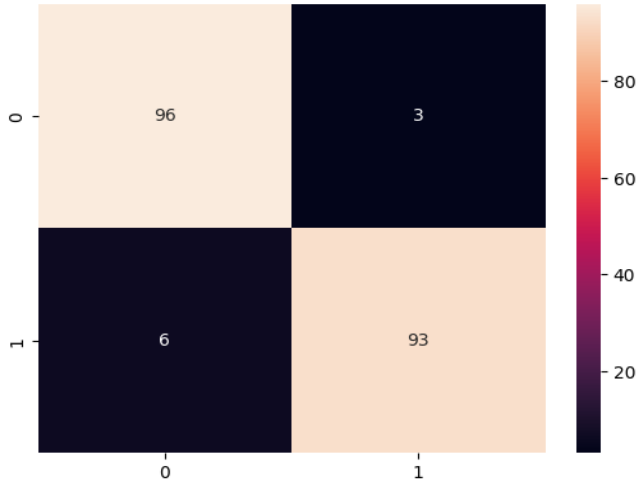

```
# 시드값 설정
np.random.seed(0)
tf.random.set_seed(0)

# 데이터 전처리
scaler = StandardScaler()
X_train2 = scaler.fit_transform(X_train2)
X_test2 = scaler.fit_transform(X_test2)

# 모형 생성
model = keras.models.Sequential()
model.add(keras.layers.Dense(64, input_shape=(X_train2.shape[1],), activation='relu'))
model.add(Dropout(0.2))
model.add(keras.layers.Dense(32, activation='relu'))
model.add(Dropout(0.2))
model.add(keras.layers.Dense(1, activation='sigmoid'))

# 학습 과정 설정
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# 모형 학습
history = model.fit(X_train2, y_train2, validation_split=0.2, epochs=100, batch_size=64, verbose=2)
```

모델링

DNN (Deep Neural Network, 심층신경망) – 결과



epoch = 20



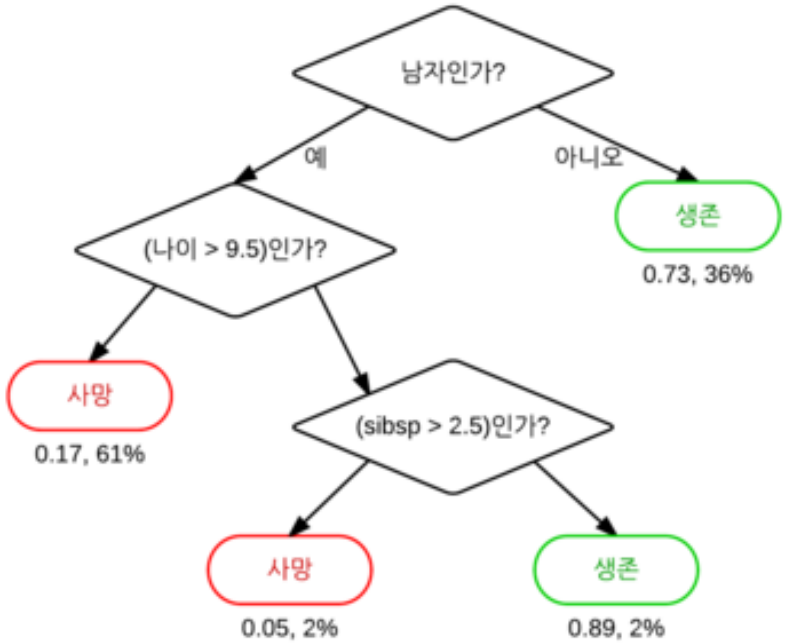
	precision	recall	f1-score	support
0	0.94	0.97	0.96	99
1	0.97	0.94	0.95	99
accuracy			0.95	198
macro avg	0.95	0.95	0.95	198
weighted avg	0.95	0.95	0.95	198

	오차	정확도
학습데이터	0.009	1.000
평가데이터	0.341	0.955

모델링

결정 트리 (decision tree learning)

몇몇 입력 변수를 바탕으로 목표 변수의 값을 예측하는 모델을 생성하는 것을 목표
하이퍼파라미터를 찾기 위해 그리드 서치를 실시, 과적합을 방지하기 위해 최적의 max_depth = 3를 찾았다.



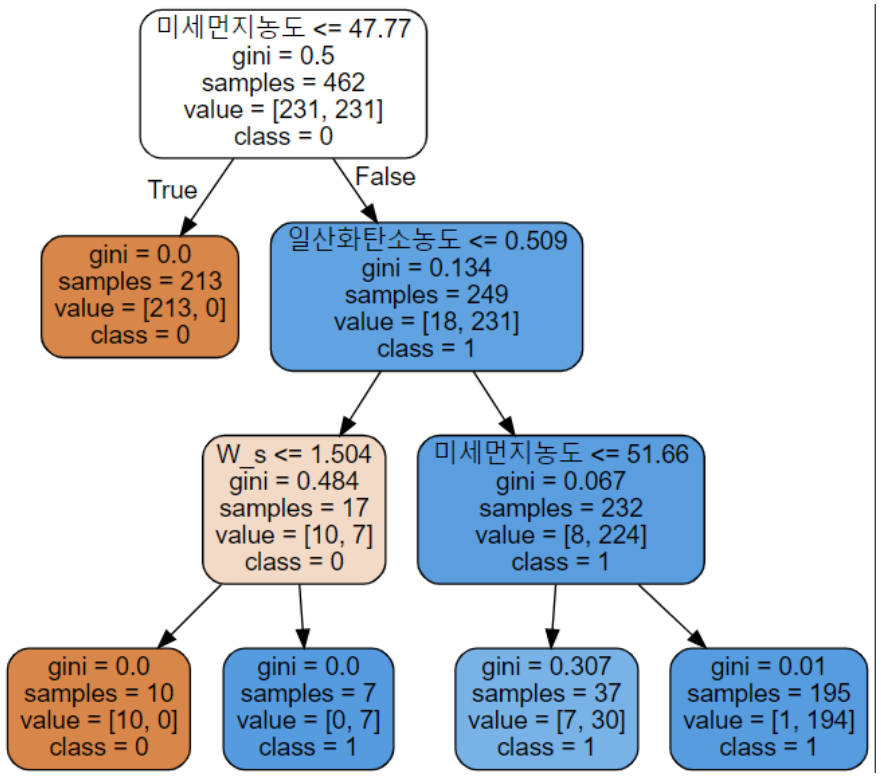

```
# 결정트리 모델 생성 및 훈련
from sklearn.tree import DecisionTreeClassifier
tree = DecisionTreeClassifier(random_state=0, max_depth=4)
tree.fit(X_train2, y_train2)
```

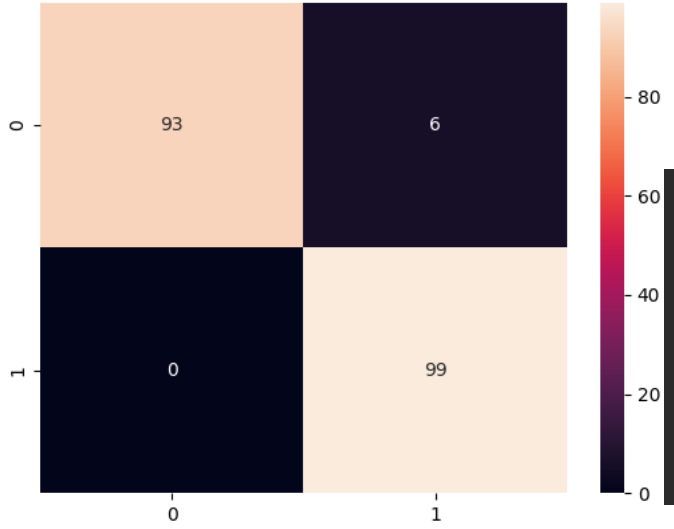
```
#max_depth (과적합 문제 해결)
from sklearn.model_selection import GridSearchCV
params = {
    'max_depth' : [2, 3, 4, 5, 6, 8, 10]
        }

grid_cv = GridSearchCV(tree, param_grid=params, scoring='accuracy', cv=5, verbose=1)
grid_cv.fit(X_train, y_train)
print('GridSearchCV 최고 평균 정확도 수치: {:.4f}'.format(grid_cv.best_score_))
print('GridSearchCV 최적 하이퍼파라미터: ', grid_cv.best_params_)
```

```
GridSearchCV 최고 평균 정확도 수치: 0.9843
GridSearchCV 최적 하이퍼파라미터:  {'max_depth': 3}
```

모델링

결정 트리 (decision tree learning) – 결과





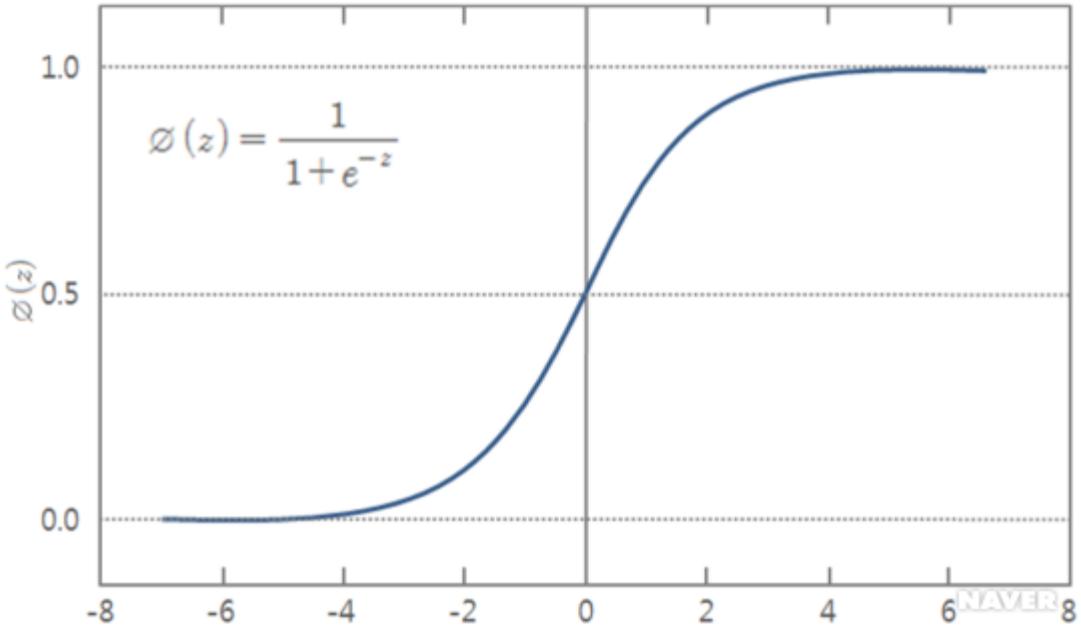
	precision	recall	f1-score	support
0	1.00	0.94	0.97	99
1	0.94	1.00	0.97	99
accuracy			0.97	198
macro avg	0.97	0.97	0.97	198
weighted avg	0.97	0.97	0.97	198

	정확도
학습데이터	0.983
평가데이터	0.970

모델링

로지스틱 회귀 (logistic regression)

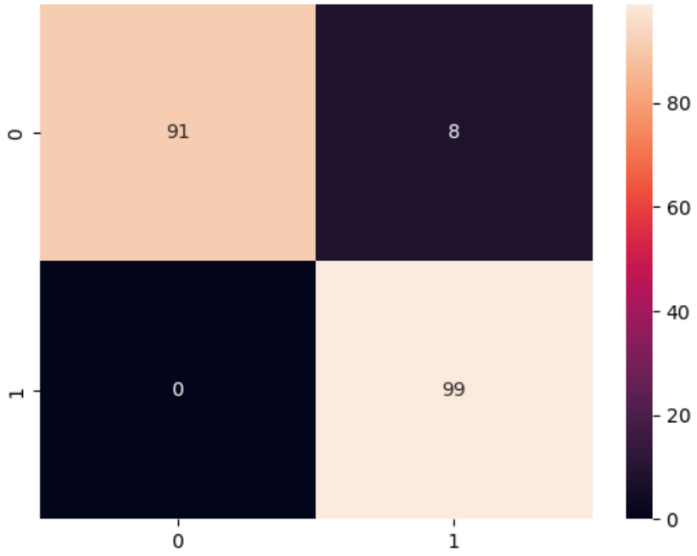
로지스틱 회귀는 예측 분석을 위한 회귀분석 중에서 특히 종속 변수가 이분형일 때 수행할 수 있는 회귀 분석 기법의 한 종류이다. 로지스틱 회귀 분석을 통해 하나의 종속 변수와 여러 독립 변수 간의 다변수 회귀 관계를 조사할 수 있다.




```
# 종속변수가 이진 모델 (binomial) -> 로지스틱 모델 사용
from sklearn.linear_model import LogisticRegression

# 로지스틱 회귀 모델 생성 (L1 정규화 적용)
# max_iter 매개변수 설정하지 않으면 수렴 실패 -> max_iter 매개변수 늘리기
logit_model = LogisticRegression(penalty='l1', solver='liblinear', random_state=100, max_iter=1000)

# 모델 학습
logit_model.fit(X_train2, y_train2)

# 모델 예측
y_pred = logit_model.predict(X_test2)
```

모델링

로지스틱 회귀(logistic regression) – 결과



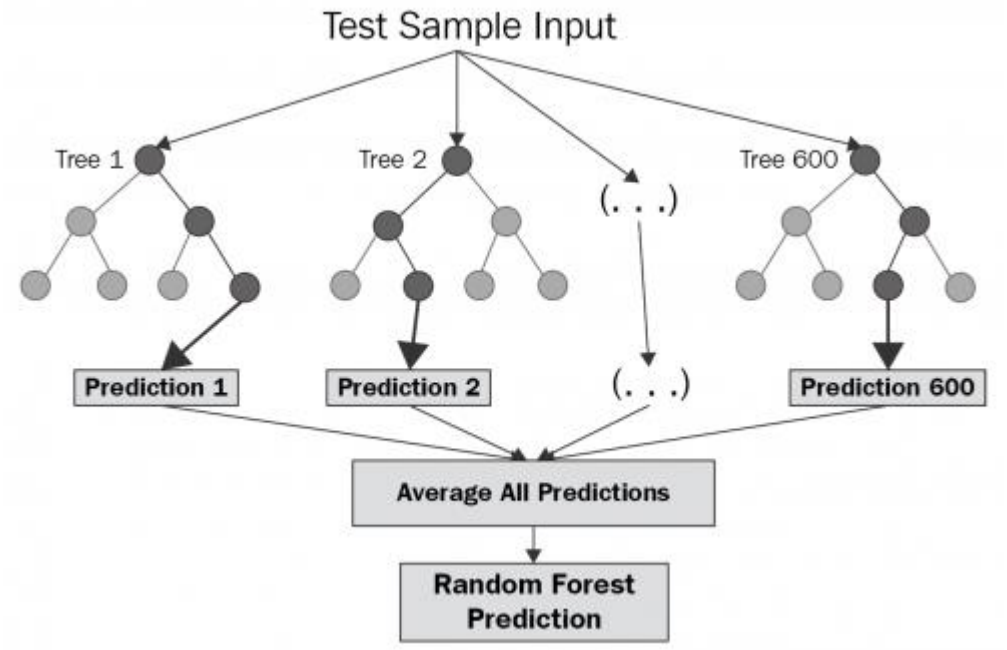
	precision	recall	f1-score	support
0	1.00	0.92	0.96	99
1	0.93	1.00	0.96	99
accuracy			0.96	198
macro avg	0.96	0.96	0.96	198
weighted avg	0.96	0.96	0.96	198

	오차	정확도
학습데이터	0.13	0.987
평가데이터	0.04	0.96

모델링

랜덤포레스트 (RandomForest)

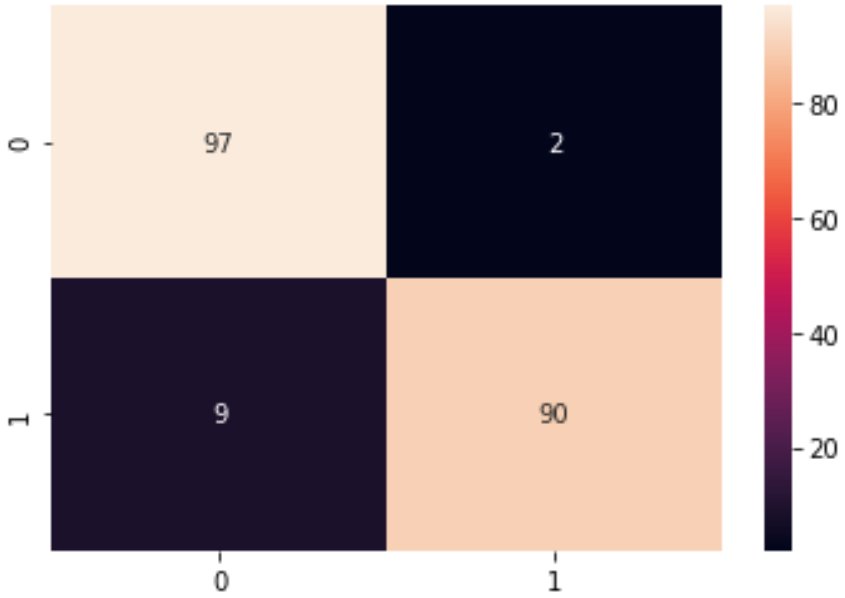
랜덤 포레스트는 다수의 결정 트리들을 학습하는 앙상블 방법이다. 랜덤 포레스트의 가장 큰 특징은 랜덤성(randomness)에 의해 트리들이 서로 조금씩 다른 특성을 갖는다는 점이다. 랜덤 학습 데이터 추출 방법을 이용한 앙상블 학습법인 배깅(bagging)과 랜덤 노드 최적화(randomized node optimization)가 자주 사용된다.




```
[ ]  model = RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
                                    max_depth=None, max_features='auto', max_leaf_nodes=None,
                                    min_impurity_decrease=0.0,
                                    min_samples_leaf=1, min_samples_split=2,
                                    min_weight_fraction_leaf=0.0, n_estimators=100,
                                    n_jobs=None, oob_score=False, random_state=None,
                                    verbose=0, warm_start=False)
```

모델링

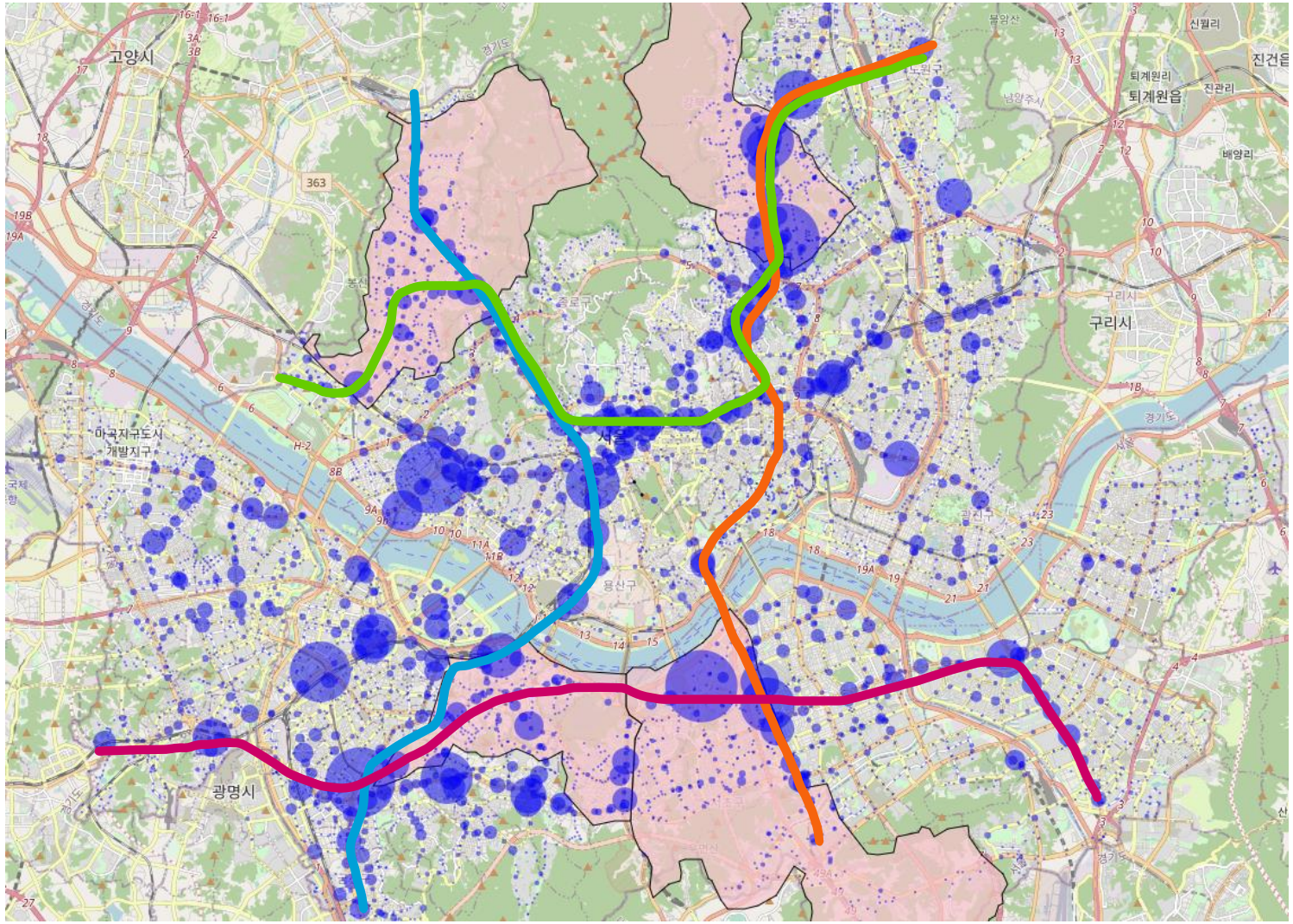
결정 트리 (decision tree learning) – 결과



	precision	recall	f1-score	support
0	0.92	0.98	0.95	99
1	0.98	0.91	0.94	99
accuracy			0.94	198
macro avg	0.95	0.94	0.94	198
weighted avg	0.95	0.94	0.94	198

	정확도
의사결정나무 10개	0.9444
의사결정나무 100개	0.9444

분석결과




```
AUC: 0.9401579586877279
Accuracy: 0.9401579586877278
Recall: 0.944309437019036
Precision: 0.9365334404498895
F1 스코어: 0.9404053645255621
```

	측정소명
서초구	180
강북구	162
동작구	153
은평구	149
강변북로	139
남산	138
동작대로	133

분석 결과 **서초 > 강북 > 동작 > 은평** 순으로 기상과 대기오염에 의한 미세먼지의 영 향을 많이 받는 것을 도출

서울시 내의 최적의 미세먼지 저감정책 효과를 나타낼 수 있는 수소버스 운용방안으로 임의의 선 위의 역을 지나는 노선추천