

Term project report

2015147533 유현석

1. 주제 선정

앞서 Term project proposal 보고서에 설명했다시피 저는 현재 강아지를 키우고 있습니다. 같이 지내고 많은 시간을 보내다 보니 자연스럽게 강아지에게 더욱 많은 관심을 두게 되었고 거기에서 예전에 보았던 사회적 문제가 생각났었습니다. 바로 유기되는 수많은 강아지와 그 강아지들이 보호소로 이동되고 대기하다가 결국 안락사하게 되는 문제입니다.

그렇다면 왜 수많은 강아지가 유기되고 안락사를 당하게 되는가 자료를 찾아보고 그 과정에서 어떠한 문제점들이 있는지 생각하게 되었습니다.

예전과 비교하면 최근 애견 및 다양한 애완동물이 사람들에게 친숙하게 다가오고 국민의 20%가 반려동물을 키운다는 사실을 접했습니다. 그런 만큼 반려동물에 대한 문제가 많이 나오고 있는데 과연 앞으로는 얼마나 많은 사람이 강아지를 키우고 또 얼마나 많은 강아지가 버려지고 안락사당할지 지금까지의 데이터를 토대로 예측을 하고 싶었습니다. 또한, 예측된 데이터를 토대로 앞에 언급했던 문제들을 해결해보고 싶었습니다.

2. 데이터 자료

해당 데이터는 직접 조사하여 엑셀에 정리한 데이터입니다. 아래의 데이터들을 이용하여 예측 데이터를 구할 것입니다. 해당 데이터들은 서울시 자료, 논문, 농림축산검역본부 등에서 조사하여 참고한 자료이고 아래 참고자료에 명시할 예정입니다.

2-1 Seoul sheet

아래의 첫 번째 시트인 Seoul은 서울시 자료를 참고하여 만든 데이터입니다. 'percent of Seoul'은 서울시 인구 중에서 애완동물을 키우는 비율입니다. 'dog percent'는 애완동물을 키우는 사람 중에서 강아지를 키우는 사람들의 비율입니다.

즉 두 개의 데이터를 종합하면 전체 서울시 인구 중에서 강아지 키우는 사람들의 비율을 구할 수 있습니다. 서울시 데이터를 선정한 이유로는 제가 살고 있는 곳이 서울이기도 하지만 시골에 내려 갈수록 키우는 강아지들을 확인하기 힘든 경우들과 대한민국의 수도인 서울이 전체 도시의 기준이 될 수 있다고 생각했기 때문입니다. 데이터는 다음과 같습니다.

<seoul sheet>

year	percent of seoul	dog percent
2014	18.8	88.9
2015	19.4	88.7
2016	19	85
2017	19.4	84.6
2018	20	84.9

2-2 All sheet

아래의 all sheet는 서울시만의 비율이 아닌 전체 대한민국의 비율을 구한 것입니다. ‘dog percent’는 전체에서 강아지를 키우는 비율이고 ‘per house’ 가구당 키우는 강아지의 숫자입니다. ‘total dog’는 전체 강아지의 숫자입니다. 해당 데이터는 다음과 같습니다.

<all sheet>

year	dog percent	per house	total dog
2006	22.1	1.66	5,988,000
2010	16.3	1.47	4,192,000
2012	16	1.38	4,001,000
2015	19.1	1.28	4,648,000
2017	24.2	1.34	6,318,000

2-3 Lost sheet

아래의 lost sheet는 유기되거나 보호소로 이동하게 되는 동물들에 대한 자료를 중심으로 두고 있습니다. ‘lost animal’은 한해 보호소로 이동하는 동물들의 숫자이고 ‘lost dog’는 그 중 강아지의 숫자, ‘percent of dog’는 이를 토대로 구한 비율입니다. ‘center number’는 보호 센터의 숫자이고 ‘re-adoption’은 재입양 비율, ‘back to home’은 원래 주인을 찾을 비율, ‘artificial- dead’는 안락사된 비율입니다. 해당 데이터는 아래와 같습니다.

<lost sheet>

year	lost animal	lost dog	percent of dog	center number	re-adoption	back to home	artificial-dead
2011	96,268	55,902	0.58069140 ₃	339	26.7	7.6	26.7
2012	99,237	59,168	0.59622922 ₉	349	27.4	8.7	24.5
2013	97,197	62,119	0.63910408 ₈	361	28.1	10.2	24.6
2014	81,147	59,180	0.72929375 ₁	343	31.4	13	22.7
2015	82,082	59,633	0.72650520 ₂	307	32	14.6	20
2016	89,732	63,602	0.70879953 ₆	281	30.4	15.2	19.9
2017	102,59 ₃	74,337	0.72458159 ₉	293	30.2	14.5	20.2

2-4 New sheet

new sheet는 특별히 수요와 공급의 원칙에 따라 과연 매년 얼마나 많은 강아지가 새로 분양되고 잃어버려지는지 보여주기 위한 자료입니다. ‘new’는 매년 새로 입양되는 강아지들의 숫자이고 ‘lost’는 매년 잃어버려지는 강아지들의 숫자입니다.

<new sheet>

year	new	lost
2015	91000	82000
2016	92000	90000
2017	105000	103000
2018	147000	121000

3. 사용한 방법론

이번 학기에 배운 주된 4가지 방법인 Optimization, Cluster Analysis, Naive Bayes, Forecasting 중에서 저는 Optimization과 Forecasting을 이용하여 문제를 해결하고자 하였습니다.

주로 사용한 방법은 Forecasting은 현재의 데이터를 가지고 미래의 데이터를 예측하기 위한 수단으로 많이 사용되었습니다. 그 과정에서 Optimization은 필수적으로 들어갔습니다. 왜냐하면, 더 정확한 예측을 위해서 오차를 최소화해야 하는데 그 과정에서 사용하기 때문입니다.

3-1 함수 import 및 Data frame 만들기

```
import pandas as pd
import numpy as np
import xlrd
import os
from statsmodels.tsa.holtwinters import ExponentialSmoothing
import matplotlib.pyplot as plt
from scipy.stats import linregress

df=pd.read_excel(r'C:\Users\HyunSeok\Desktop\air term.xlsx',sheet_name='seoul')
df1=pd.read_excel(r'C:\Users\HyunSeok\Desktop\air term.xlsx',sheet_name='all')
df2=pd.read_excel(r'C:\Users\HyunSeok\Desktop\air term.xlsx',sheet_name='lost')
df3=pd.read_excel(r'C:\Users\HyunSeok\Desktop\air term.xlsx',sheet_name='new')
```

3-2 1번 Sheet (Seoul sheet) 예측 데이터 만들기

이중 평활 예측법인 Holt's Trend-corrected Exponential Smoothing을 이용하여 예측을 하였습니다. Holt's Trend-corrected Exponential Smoothing을 사용한 이유는 이번 과제에서 원하는 자료는 지금까지의 trend를 기반으로 앞으로 과연 어떻게 변화가 생길지를 알아보고 싶었기 때문에 level과 trend가 존재하는 해당 방법을 사용하였습니다. 이는 앞으로 예측하는 모든 데이터에 사용될 방법입니다.

<Data 1 예측하기>

```
data1=df['percent of seoul']
data2=df['dog percent']

hz= ExponentialSmoothing(data1,trend='add').fit()
C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\qual
loc = initial_p >= ub

forecasting=hz.forecast(7)

for i in range(5,12):
    df.loc[i]=np.nan
for i in range(5,12):
    df['percent of seoul'][i]=forecasting[i]
```

<Data 2 예측하기>

```
hz= ExponentialSmoothing(data2,trend='add').fit()
C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\qual
loc = initial_p >= ub

forecasting=hz.forecast(7)
for i in range(5,12):
    df['dog percent'][i]=forecasting[i]
    df['year'][i]=2014+i
```

위의 그림은 첫 번째 시트의 column을 각각 data 1, data 2에 저장하였고 Exponential Smoothing 함수를 이용하여 오차를 Optimization 해주었습니다. 그리고 7년 뒤까지 예측해보았습니다.

다음 그림은 엑셀을 통해서 읽어온 자료를 data frame으로 저장하여 예측하기 전, 원래의 자료와 예측한 데이터까지 저장한 data frame을 비교하였습니다.

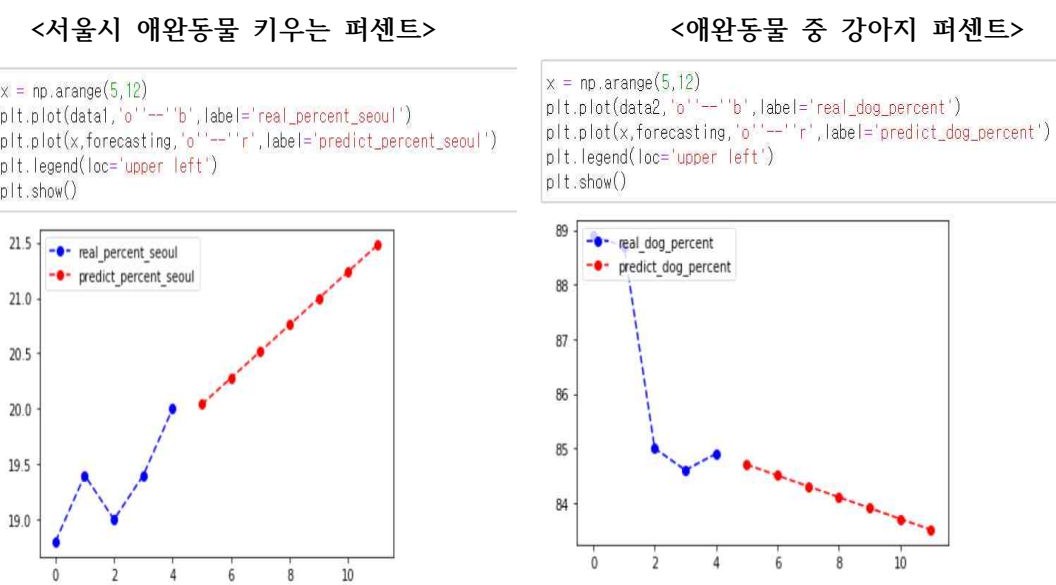
<예측하기 전 Data Frame>

	year	percent of seoul	dog percent
0	2014	18.8	88.9
1	2015	19.4	88.7
2	2016	19.0	85.0
3	2017	19.4	84.6
4	2018	20.0	84.9

<예측하기 후 Data Frame>

	year	percent of seoul	dog percent
0	2014.0	18.800000	88.900000
1	2015.0	19.400000	88.700000
2	2016.0	19.000000	85.000000
3	2017.0	19.400000	84.600000
4	2018.0	20.000000	84.900000
5	2019.0	20.039412	84.701209
6	2020.0	20.279198	84.502418
7	2021.0	20.518985	84.303626
8	2022.0	20.758772	84.104835
9	2023.0	20.998558	83.906044
10	2024.0	21.238345	83.707253
11	2025.0	21.478131	83.508462

아래의 그래프는 위에서 얻은 데이터를 토대로 각각의 컬럼에 해당하는 data 값들을 보기 편하게 그래프로 나타내었습니다. 파란색은 기존의 데이터고 빨간색은 예측 데이터입니다. 그래프를 통해서 변화를 눈에 보기 쉽게 하였습니다.



3-2 2번 Sheet (All sheet) 예측 데이터 만들기

아래의 그림은 두 번째 시트인 All sheet의 3개의 column을 각각 data 1, data 2, data 3에 저장하였고 Exponential Smoothing 함수를 이용하여 오차를 Optimization 해주었습니다. 그리고 8년 뒤까지 예측해보았습니다. 아래의 Data Frame을 보시면 알겠지만, 이번 자료는 1년 주기로 data가 있는 게 아니라 2, 3년 주기로 data가 있어 저 역시 같은 방식으로 2, 3년 주기로 2025년까지의 예측 data를 구하였습니다.

<Data 1 예측하기>	<Data 2 예측하기>	<Data 3 예측하기>
<pre>data1=df1['dog percent'] data2=df1['per house'] data3=df1['total dog']</pre>		
<pre>hz= ExponentialSmoothing(data1,trend='add').fit() C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\ qual loc = initial_p >= ub C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\ heck mle_retvals. ConvergenceWarning)</pre>	<pre>hz= ExponentialSmoothing(data2,trend='add').fit() C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\ qual loc = initial_p >= ub</pre>	<pre>hz= ExponentialSmoothing(data3,trend='add').fit() C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\ qual loc = initial_p >= ub C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\ heck mle_retvals. ConvergenceWarning)</pre>
<pre>forecasting=hz.forecast(3)</pre>	<pre>forecasting=hz.forecast(3)</pre>	<pre>forecasting=hz.forecast(3)</pre>
<pre>for i in range(5,8): df1.loc[i]=np.nan for i in range(5,8): df1['dog percent'][i]=forecasting[i] df1['year'][5]=2020 df1['year'][6]=2023 df1['year'][7]=2025</pre>	<pre>for i in range(5,8): df1['per house'][i]=forecasting[i]</pre>	<pre>for i in range(5,8): df1['total dog'][i]=forecasting[i]</pre>

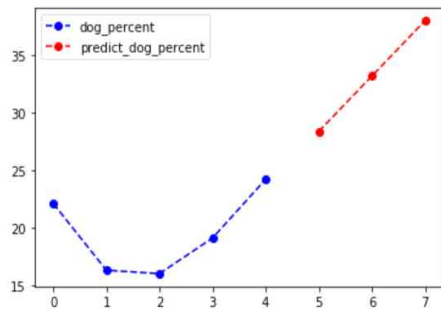
아래의 data frame들 역시 1번 시트와 마찬가지로 예측된 데이터를 구하기 전의 data frame과 2025년까지의 예측된 데이터가 있는 data frame으로 나타내었습니다.

<Data Frame 1 예측 전 >	<Data Frame 1 예측 후>
df1	df1
year dog percent per house total dog	year dog percent per house total dog
0 2006 22.1 1.66 5988000	0 2006.0 22.100000 1.660000 5.988000e+06
1 2010 16.3 1.47 4192000	1 2010.0 16.300000 1.470000 4.192000e+06
2 2012 16.0 1.38 4001000	2 2012.0 16.000000 1.380000 4.001000e+06
3 2015 19.1 1.28 4648000	3 2015.0 19.100000 1.280000 4.648000e+06
4 2017 24.2 1.34 6318000	4 2017.0 24.200000 1.340000 6.318000e+06
	5 2020.0 28.363517 1.226252 7.521178e+06
	6 2023.0 33.168562 1.175222 8.970838e+06
	7 2025.0 37.973607 1.124193 1.042050e+07

아래의 3개의 그래프는 위에서 얻은 데이터를 토대로 각각의 컬럼에 해당하는 data 값들을 보기 편하게 그래프로 나타내었습니다. 마찬가지로 파란색은 기존의 데이터고 빨간색은 예측 데이터입니다. 그래프를 통해서 변화를 눈에 보기 쉽게 하였습니다.

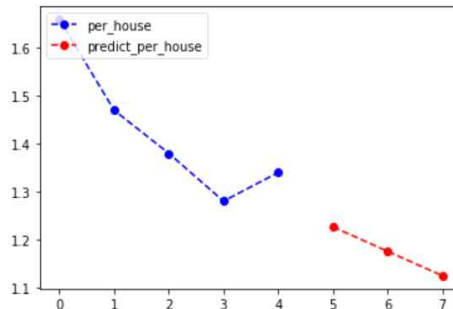
<전체 인구 중 애완동물 키우는 퍼센트>

```
x = np.arange(5,8)
plt.plot(data1, 'o'-'-' 'b', label='dog_percent')
plt.plot(x, forecasting, 'o'-'-' 'r', label='predict_dog_percent')
plt.legend(loc='upper left')
plt.show()
```



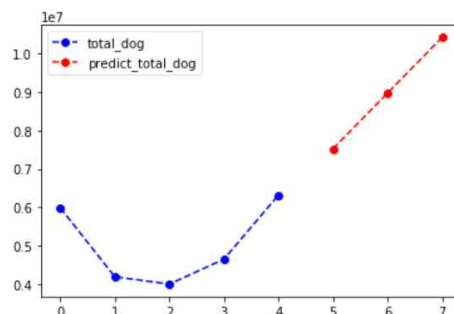
<한 가구 당 키우는 강아지의 평균 숫자>

```
x = np.arange(5,8)
plt.plot(data2, 'o'-'-' 'b', label='per_house')
plt.plot(x, forecasting, 'o'-'-' 'r', label='predict_per_house')
plt.legend(loc='upper left')
plt.show()
```



<전체 강아지의 개체 수>

```
x = np.arange(5,8)
plt.plot(data3, 'o'-'-' 'b', label='total_dog')
plt.plot(x, forecasting, 'o'-'-' 'r', label='predict_total_dog')
plt.legend(loc='upper left')
plt.show()
```



3-3 3번 Sheet (lost sheet) 예측 데이터 만들기

아래의 그림은 세 번째 시트인 lost sheet의 7개의 column을 각각 data 1, data 2, data 3 data 7에 저장하였고 Exponential Smoothing 함수를 이용하여 오차를 Optimization 해주었습니다. 그리고 마찬가지로 8년 뒤인 2025년까지 예측해보았습니다. 그전까지는 column의 숫자가 많지 않아서 한 번에 하나씩 구해 계산했지만, 이번에는 7개나 되기 때문에 7개의 변수를 이용하여 Exponential Smoothing 함수를 계산해주었습니다.

<Data 1 ~ Data 7 예측하기>

```
data1=df2['lost animal']
data2=df2['lost dog']
data3=df2['percent of dog']
data4=df2['center number']
data5=df2['re-adoption']
data6=df2['back to home']
data7=df2['artificial- dead']
```

```
hz1= ExponentialSmoothing(data1,trend='add').fit()
hz2= ExponentialSmoothing(data2,trend='add').fit()
hz3= ExponentialSmoothing(data3,trend='add').fit()
hz4= ExponentialSmoothing(data4,trend='add').fit()
hz5= ExponentialSmoothing(data5,trend='add').fit()
hz6= ExponentialSmoothing(data6,trend='add').fit()
hz7= ExponentialSmoothing(data7,trend='add').fit()
```

<8년 뒤 Data까지 예측하기>

```
forecasting1=hz1.forecast(8)
forecasting2=hz2.forecast(8)
forecasting3=hz3.forecast(8)
forecasting4=hz4.forecast(8)
forecasting5=hz5.forecast(8)
forecasting6=hz6.forecast(8)
forecasting7=hz7.forecast(8)
```

```
for i in range(7,15):
    df2.loc[i]=np.nan
for i in range(7,15):
    df2['year'][i]=2011+i
    df2['lost animal'][i]=forecasting1[i]
    df2['lost dog'][i]=forecasting2[i]
    df2['percent of dog'][i]=forecasting3[i]
    df2['center number'][i]=forecasting4[i]
    df2['re-adoption'][i]=forecasting5[i]
    df2['back to home'][i]=forecasting6[i]
    df2['artificial- dead'][i]=forecasting7[i]
```

아래의 data frame들 역시 1, 2번 시트와 마찬가지로 예측된 데이터를 구하기 전의 data frame과 2025년까지의 예측된 데이터가 있는 data frame으로 나타내었습니다. 7 개의 데이터들을 각각 합쳐서 나타내었습니다.

<예측하기 전 Data Frame>

df2

	year	lost animal	lost dog	percent of dog	center number	re-adoption	back to home	artificial- dead
0	2011	96268	55902	0.580691	339	26.7	7.6	26.7
1	2012	99237	59168	0.596229	349	27.4	8.7	24.5
2	2013	97197	62119	0.639104	361	28.1	10.2	24.6
3	2014	81147	59180	0.729294	343	31.4	13.0	22.7
4	2015	82082	59633	0.726505	307	32.0	14.6	20.0
5	2016	89732	63602	0.708800	281	30.4	15.2	19.9
6	2017	102593	74337	0.724582	293	30.2	14.5	20.2

<예측한 후의 Data Frame>

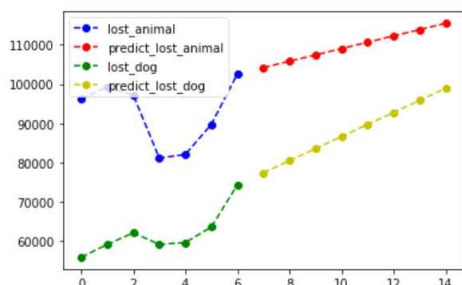
	year	lost animal	lost dog	percent of dog	center number	re-adoption	back to home	artificial- dead
0	2011.0	96268.000000	55902.000000	0.580691	339.0	26.700000	7.600000	26.700000
1	2012.0	99237.000000	59168.000000	0.596229	349.0	27.400000	8.700000	24.500000
2	2013.0	97197.000000	62119.000000	0.639104	361.0	28.100000	10.200000	24.600000
3	2014.0	81147.000000	59180.000000	0.729294	343.0	31.400000	13.000000	22.700000
4	2015.0	82082.000000	59633.000000	0.726505	307.0	32.000000	14.600000	20.000000
5	2016.0	89732.000000	63602.000000	0.708800	281.0	30.400000	15.200000	19.900000
6	2017.0	102593.000000	74337.000000	0.724582	293.0	30.200000	14.500000	20.200000
7	2018.0	104202.057403	77409.487021	0.748563	305.0	32.371362	15.650000	18.226553
8	2019.0	105811.114806	80481.974042	0.772545	317.0	33.099910	16.800000	17.252683
9	2020.0	107420.172209	83554.461063	0.796527	329.0	33.828458	17.950000	16.278813
10	2021.0	109029.229613	86626.948085	0.820508	341.0	34.557006	19.100000	15.304944
11	2022.0	110638.287016	89699.435106	0.844490	353.0	35.285554	20.250000	14.331074
12	2023.0	112247.344419	92771.922127	0.868472	365.0	36.014102	21.399999	13.357205
13	2024.0	113856.401822	95844.409148	0.892453	377.0	36.742650	22.549999	12.383335
14	2025.0	115465.459225	98916.896169	0.916435	389.0	37.471198	23.699999	11.409465

아래의 4개의 그래프는 위에서 얻은 데이터를 토대로 각각의 column에 해당하는 data 값들을 보기 편하게 그래프로 나타내었습니다. 마찬가지로 파란색은 기존의 데이터고 빨간색은 예측 데이터입니다. 연관성이 있는 데이터들은 하나의 그래프에 나타내었습니다.

첫 번째 그래프는 잃어버리는 동물과 강아지의 숫자이기 때문에 하나로 나타내었습니다. 파란색과 빨간색과 마찬가지로 초록색이 기존의 데이터, 노란색이 예측 데이터입니다.

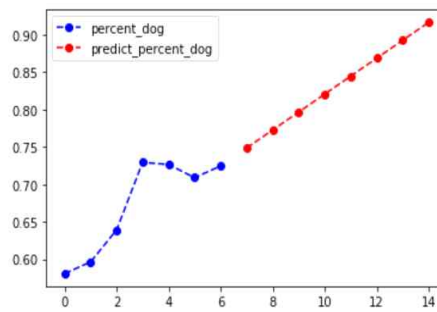
<잃어버린 동물과 강아지 숫자>

```
x = np.arange(7,15)
plt.plot(data1, 'o'--'b', label='lost_animal')
plt.plot(x, forecasting1, 'o'--'r', label='predict_lost_animal')
plt.plot(data2, 'o'--'g', label='lost_dog')
plt.plot(x, forecasting2, 'o'--'y', label='predict_lost_dog')
plt.legend(loc='upper left')
plt.show()
```



<잃어버리는 전체 동물 중 강아지 비율>

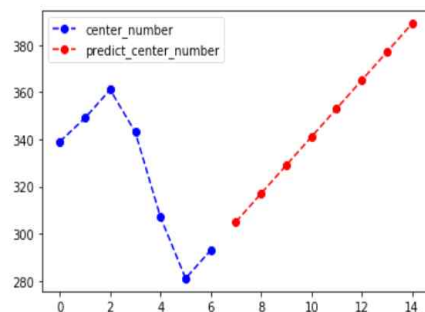
```
x = np.arange(7,15)
plt.plot(data3, 'o'--'b', label='percent_dog')
plt.plot(x, forecasting3, 'o'--'r', label='predict_percent_dog')
plt.legend(loc='upper left')
plt.show()
```



4번 그래프는 잃어버리는 동물이 보호 센터로 보내져서 직면하게 되는 미래들의 유형들의 비율을 나타내었습니다. 크게 3가지 유형인데 원래 주인을 찾거나, 새로운 주인을 찾거나, 안락사되는 미래입니다. 파란색과 빨간색과 마찬가지로 초록색이 기존의 데이터, 노란색이 예측 데이터, 검은색이 기존의 데이터, 하늘색이 예측 데이터입니다.

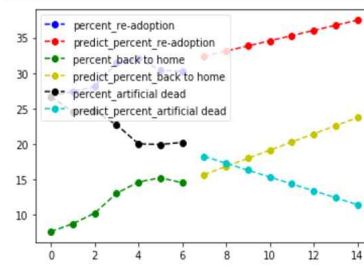
<동물 보호 센터의 숫자>

```
x = np.arange(7,15)
plt.plot(data4, 'o'-'-'-'b', label='center_number')
plt.plot(x, forecasting4, 'o'-'-'-'r', label='predict_center_number')
plt.legend(loc='upper left')
plt.show()
```



<보호소의 동물의 직면하는 미래 비율>

```
x = np.arange(7,15)
plt.plot(data5, 'o'-'-'-'b', label='percent_re-adoption')
plt.plot(x, forecasting5, 'o'-'-'-'r', label='predict_percent_re-adoption')
plt.plot(data6, 'o'-'-'-'g', label='percent_back to home')
plt.plot(x, forecasting6, 'o'-'-'-'y', label='predict_percent_back to home')
plt.plot(data7, 'o'-'-'-'k', label='percent_artificial dead')
plt.plot(x, forecasting7, 'o'-'-'-'c', label='predict_percent_artificial dead')
plt.legend(loc='upper left')
plt.show()
```



3-4 4번 Sheet (New sheet) 예측 데이터 만들기

아래의 그림은 네 번째 시트인 New sheet의 2개의 column을 각각 data 1, data 2에 저장하였고 Exponential Smoothing 함수를 이용하여 오차를 Optimization 해주었습니다. 그리고 마찬가지로 7년 뒤인 2025년까지 예측해보았습니다. 이번에도 세 번째 시트와 마찬가지로 2개의 함수를 동시에 이용하여 계산하였습니다.

<8년뒤 Data 1, Data 2 예측하기>

```
data1=df3['new']
data2=df3['lost']

hz1= ExponentialSmoothing(data1,trend='add').fit()
hz2= ExponentialSmoothing(data2,trend='add').fit()

C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\
qual
loc = initial_p >= ub
C:\ProgramData\Anaconda3\lib\site-packages\statsmodels\
heck_mle_retvals
ConvergenceWarning)

forecasting1=hz1.forecast(7)
forecasting2=hz2.forecast(7)

for i in range(4,11):
    df3.loc[i]=np.nan
for i in range(4,11):
    df3['year'][i]=2015+i
    df3['new'][i]=forecasting1[i]
    df3['lost'][i]=forecasting2[i]
```

아래의 data frame들 역시 앞선 시트들과 마찬가지로 예측된 데이터를 구하기 전의 data frame과 2025년까지의 예측된 데이터가 있는 data frame으로 나타내었습니다.

<예측하기 전 Data Frame>

df3

	year	new	lost
0	2015	91000	82000
1	2016	92000	90000
2	2017	105000	103000
3	2018	147000	121000

<예측 후 Data Frame>

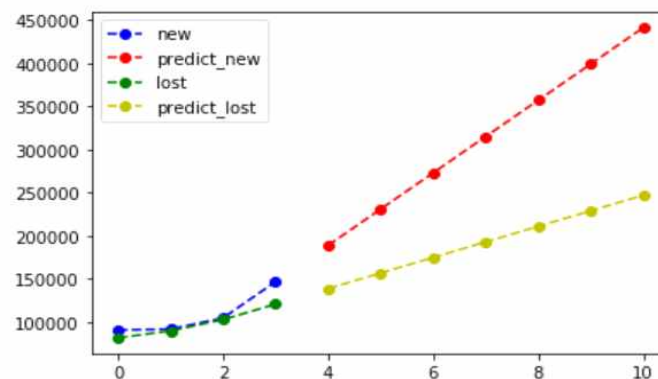
df3

	year	new	lost
0	2015.0	91000.00000	82000.00000
1	2016.0	92000.00000	90000.00000
2	2017.0	105000.00000	103000.00000
3	2018.0	147000.00000	121000.00000
4	2019.0	188992.49928	138999.663496
5	2020.0	230987.89891	156999.479718
6	2021.0	272983.29854	174999.295941
7	2022.0	314978.69817	192999.112164
8	2023.0	356974.09780	210998.928386
9	2024.0	398969.49743	228998.744609
10	2025.0	440964.89706	246998.560832

아래의 그래프는 2개의 데이터인 새로 키우게 되는 강아지의 숫자와 잃어버려지는 강아지들의 숫자를 나타냈습니다. 두 개의 데이터는 연관성이 있다고 생각이 들어 하나의 그래프에 나타내었습니다. 마찬가지로 파란색은 기존의 데이터고 빨간색은 예측 데이터, 초록색이 기존의 데이터, 노란색이 예측 데이터입니다.

<새로 키우는 강아지 숫자와 잃어버리는 강아지 숫자>

```
x = np.arange(4,11)
plt.plot(data1,'o'--'b',label='new')
plt.plot(x,forecasting1,'o'--'r',label='predict_new')
plt.plot(data2,'o'--'g',label='lost')
plt.plot(x,forecasting2,'o'--'y',label='predict_lost')
plt.legend(loc='upper left')
plt.show()
```



3-5 각각의 만든 예측 데이터를 가지고 필요한 자료 구하기

각각의 완성된 예측데이터를 가지고 이제 비교 및 분석에 필요한 자료를 구할 수 있게 됩니다.

다음의 식은 안락사된 비율을 가지고 실제 얼마나 많은 강아지들이 안락사가 되는지 구체적인 숫자를 구할 수 있게 됩니다. 그 데이터 역시 해당 Data Frame에 추가하였습니다.

```
df2['number of dog(artificial dead)']=df2['lost dog']/100  
df2['number of dog(artificial dead)']=df2['number of dog(artificial dead)']*df2['artificial- dead']
```

<해당 식의 결과물>

e	artificial- dead	number of dog(artificial dead)
0	26.700000	14925.834000
0	24.500000	14496.160000
0	24.600000	15281.274000
0	22.700000	13433.860000
0	20.000000	11926.600000
0	19.900000	12656.798000
0	20.200000	15016.074000
0	18.226553	14109.080796
0	17.252683	13885.299782
0	16.278813	13601.674735
0	15.304944	13258.205654

이를 통해 얻은 결과들을 가지고 아래의 결과 도출과정에서 자세히 설명하도록 하겠습니다.

4. 결과 도출 과정

4-1 1번과 2번 시트로 알아본 도출한 결과

1번 시트(서울시의 예측결과) 결과를 통해 얻을 수 있는 점은 앞으로 서울에 있는 사람들은 애완동물을 더욱 많이 키우게 될 것이고 그 중, 가장 많은 비율을 차지했던 강아지 비율이 낮아지는 것을 보면 다양하고 색다른 애완동물들이 많아지는 것을 알 수 있습니다. 이는 앞으로의 애완동물 정책이나 보호소 및 비중을 강아지나 고양이가 아닌 다른 이색 동물에도 신경을 써야 한다는 사실을 알 수 있습니다.

이와 같은 주장은 2번 시트(전체를 대상으로 한 예측결과)에서도 찾아볼 수 있습니다. 전체 인구로 봤을 때는 가정에서 키우는 강아지의 비중은 늘어나지만, 가구당 키우는 강아지의 숫자는 줄고 있습니다. 이는 다양한 애완동물의 유입이라고도 볼 수 있습니다. 하지만 다른 이유를 생각해보면 점점 늘어나는 1인 가구에서 키우는 동물의 숫자가 한계가 있기 때문일 수도 있습니다. 하지만 둘 모두에서 찾아볼 수 있는 점은 결과적인 총 강아지 숫자는 증가할 수밖에 없다는 것이고 이는 더욱 많은 애견 관련 문제가 일어날 수밖에 없다는 점입니다.

<1번 시트 최종 data frame>

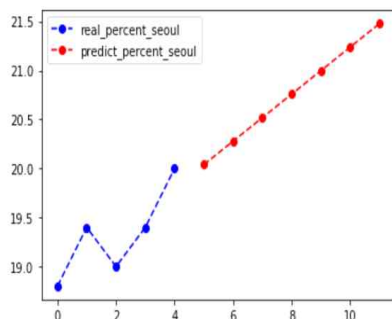
	year	percent of seoul	dog percent
0	2014.0	18.800000	88.900000
1	2015.0	19.400000	88.700000
2	2016.0	19.000000	85.000000
3	2017.0	19.400000	84.600000
4	2018.0	20.000000	84.900000
5	2019.0	20.039412	84.701209
6	2020.0	20.279198	84.502418
7	2021.0	20.518985	84.303626
8	2022.0	20.758772	84.104835
9	2023.0	20.998558	83.906044
10	2024.0	21.238345	83.707253
11	2025.0	21.478131	83.508462

<2번 시트 최종 data frame>

	year	dog percent	per house	total dog
0	2006.0	22.100000	1.660000	5.988000e+06
1	2010.0	16.300000	1.470000	4.192000e+06
2	2012.0	16.000000	1.380000	4.001000e+06
3	2015.0	19.100000	1.280000	4.648000e+06
4	2017.0	24.200000	1.340000	6.318000e+06
5	2020.0	28.363517	1.226252	7.521178e+06
6	2023.0	33.168562	1.175222	8.970838e+06
7	2025.0	37.973607	1.124193	1.042050e+07

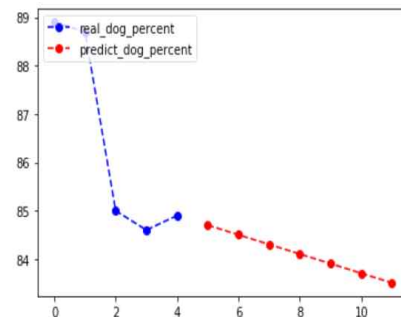
<서울시 애완동물 키우는 퍼센트>

```
x = np.arange(5,12)
plt.plot(data1, 'o'-'-' 'b', label='real_percent_seoul')
plt.plot(x, forecasting, 'o'-'-' 'r', label='predict_percent_seoul')
plt.legend(loc='upper left')
plt.show()
```



<애완동물 중 강아지 퍼센트>

```
x = np.arange(5,12)
plt.plot(data2, 'o'-'-' 'b', label='real_dog_percent')
plt.plot(x, forecasting, 'o'-'-' 'r', label='predict_dog_percent')
plt.legend(loc='upper left')
plt.show()
```



4-2 3번과 4번 시트로 알아본 도출한 결과

3번 시트는 잃어버린 동물들에 대한 데이터를 가지고 있습니다. 이는 매해 잃어버려지는 동물의 숫자와 그 중, 강아지 숫자, 그리고 앞서 설명한 것처럼 센터의 숫자, 재입양, 주인 찾기, 안락사의 비율을 가지고 있습니다. 먼저 센터의 숫자를 보면 매해 줄어들다가 최근 다시 늘어나는 추세를 가지고 있습니다.

잃어버려지는 동물의 숫자 및 강아지의 비율은 계속해서 늘어나고 있습니다. 하지만 이러한 데이터 추세는 한가지 변수를 고려하지 않았습니다. 바로 올해부터 실행된 의무적으로 반려견을 등록해야 하는 정책이 실행된 것입니다. 이로 인해 변화되는 데이터는 아직 구할 수 없어 반영하지 못하였지만, 이러한 정책으로 인해 back to home(집으로 돌아가는 비율)에서의 증가 및 보호소로 이동되는 동물 중 절대 다수를 차지하는 강아지의 비율이 낮아 질거라고 예상할 수 있습니다.

강아지가 주로 잃어버려지는 이유로는 많이 키우는 이유도 있겠지만, 산책 및 밖에 돌아다닐 기회가 다른 동물에 비해 많기 때문이라고 생각합니다. 이에 해당 정책은 강아지로 하여금 집으로 돌아갈 수 있는 더 많은 기회를 제공해 긍정적인 방향을 제공할 거 같습니다.

<3번째 시트의 최종 Data Frame>

df2								
	year	lost animal	lost dog	percent of dog	center number	re-adoption	back to home	artificial- dead
0	2011.0	96268.000000	55902.000000	0.580691	339.0	26.700000	7.600000	26.700000
1	2012.0	99237.000000	59168.000000	0.596229	349.0	27.400000	8.700000	24.500000
2	2013.0	97197.000000	62119.000000	0.639104	361.0	28.100000	10.200000	24.600000
3	2014.0	81147.000000	59180.000000	0.729294	343.0	31.400000	13.000000	22.700000
4	2015.0	82082.000000	59633.000000	0.726505	307.0	32.000000	14.600000	20.000000
5	2016.0	89732.000000	63602.000000	0.708800	281.0	30.400000	15.200000	19.900000
6	2017.0	102593.000000	74337.000000	0.724582	293.0	30.200000	14.500000	20.200000
7	2018.0	104202.057403	77409.487021	0.748563	305.0	32.371362	15.650000	18.226553
8	2019.0	105811.114806	80481.974042	0.772545	317.0	33.099910	16.800000	17.252683
9	2020.0	107420.172209	83554.461063	0.796527	329.0	33.828458	17.950000	16.278813
10	2021.0	109029.229613	86626.948085	0.820508	341.0	34.557006	19.100000	15.304944
11	2022.0	110638.287016	89699.435106	0.844490	353.0	35.285554	20.250000	14.331074
12	2023.0	112247.344419	92771.922127	0.868472	365.0	36.014102	21.399999	13.357205
13	2024.0	113856.401822	95844.409148	0.892453	377.0	36.742650	22.549999	12.383335
14	2025.0	115465.459225	98916.896169	0.916435	389.0	37.471198	23.699999	11.409465

아래의 데이터는 위에 data frame에서 수치를 이용하여 한해에 안락사되는 강아지의 구체적인 숫자를 구해봤습니다. 매년 안락사되는 강아지의 비율은 줄어들고 있습니다. 그러나 그 숫자는 늘어나고 있습니다. 이러한 문제를 해결하기 위해서는 입양률을 늘

리고 집으로 돌아가는 비율을 높이는 방법이 있습니다. 집으로 돌아가는 방법은 앞서 말했다시피 정책으로 인해 긍정적인 방향으로 전환 될 거 같습니다. 그렇다면 입양률을 높일 수 있는 방법으론 무엇이 있을까 생각해봤습니다. 일단 새로 강아지를 키우는 인구는 4번 시트를 참고하면 점점 늘어남을 알 수 있습니다. 그리고 그런 사람들은 공통적으로, 어린 1살 이하의 강아지를 선호하고 있습니다. 이는 애견 판매점에서도 어린 강아지들 밖에 없다는 점과 유사합니다. 그렇다면 입양률을 높이기 위해서는 철저한 관리와 인식의 변화를 통해서 첫 번째 강아지가 아닌 경험이 있는 애견인들을 대상으로 어필하는 점이 중요하다고 생각합니다.

<안락사 당하는 강아지 숫자>

```
df2['number of dog(artificial dead)']
```

0	14925.834000
1	14496.160000
2	15281.274000
3	13433.860000
4	11926.600000
5	12656.798000
6	15016.074000
7	14109.080796
8	13885.299782
9	13601.674735
10	13258.205654
11	12854.892540
12	12391.735391
13	11868.734209
14	11285.888993

<4번째 시트의 최종 Data Frame>

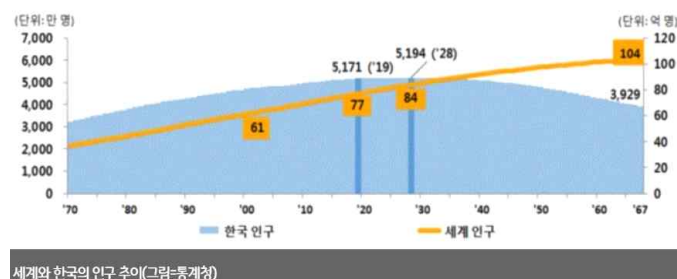
```
df3
```

	year	new	lost
0	2015.0	91000.000000	82000.000000
1	2016.0	92000.000000	90000.000000
2	2017.0	105000.000000	103000.000000
3	2018.0	147000.000000	121000.000000
4	2019.0	188992.49928	138999.663496
5	2020.0	230987.89891	156999.479718
6	2021.0	272983.29854	174999.295941
7	2022.0	314978.69817	192999.112164
8	2023.0	356974.09780	210998.928386
9	2024.0	398969.49743	228998.744609
10	2025.0	440964.89706	246998.560832

4-3 4번 시트로 알아본 도출한 결과

4번 시트인 새로 강아지를 기르는 숫자와 잃어버리는 숫자에서 데이터의 한계가 있다는 점을 알아냈습니다. 그 원인은 바로 인구의 숫자에 있습니다. 아무리 강아지나 다른 애완동물에 관한 관심이 늘어나더라도 수용치의 한계는 있을 수 밖에 없습니다. 더욱이 대한민국처럼 인구가 점점 줄어든 수세를 보이는 나라에서는 애견인의 비율이 일정부분을 넘어설 수 없습니다. 그러한 문제점들 때문에 이번 예측조사에서는 보다면 미래까지가 아닌 5년 뒤까지만 예상하였습니다.

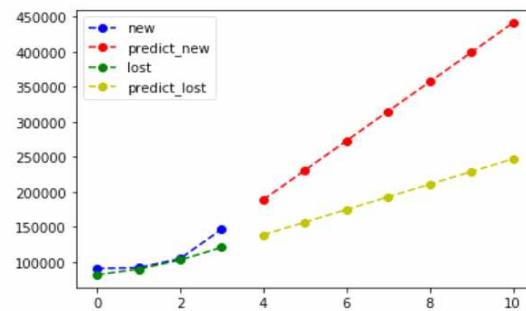
<세계와 한국의 인구 추이>



이처럼 통계청에 자료에 따르면 대한민국의 인구는 2025년까지가 제일 많을 것으로 예측되고 있습니다. 그렇기 때문에, 아래와 같이 저희 모든 예측통계 자료는 2025년까지만을 기준 삼았습니다.

<새로 키우는 강아지 숫자와 잃어버리는 강아지 숫자>

```
x = np.arange(4,11)
plt.plot(data1,'o' '--' 'b',label='new')
plt.plot(x,forecasting1,'o' '--' 'r',label='predict_new')
plt.plot(data2,'o' '--' 'g',label='lost')
plt.plot(x,forecasting2,'o' '--' 'y',label='predict_lost')
plt.legend(loc='upper left')
plt.show()
```



5. 참고자료

http://m.blog.daum.net/_blog/_m/articleView.do?blogid=0Ez8g&artcleno=11310986 (서울시 자료 인용)

<http://repository.krei.re.kr/bitstream/2018.oak/22330/1/%EB%B0%98%EB%A0%A4%EB%8F%99%EB%AC%BC%20%EC%97%B0%EA%B4%80%EC%82%B0%EC%97%85%20%EB%B0%9C%EC%A0%84%EB%B0%A9%EC%95%88%20%EC%97%B0%EA%B5%AC.pdf> (한국농촌 경제연구원 논문)

<http://www.dailyvet.co.kr/news/animalwelfare/96920> (농림축산검역본부 자료 인용)

http://stat.seoul.go.kr/jsp/WWS8/webjin_view.jsp?wj_id=118 (서울시 자료 인용)

<http://www.bigtanews.co.kr/news/articleView.html?idxno=4372> (통계청 자료 인용)