

데이터사이언스 창업전략

주제 : 달러 가격을 예측해보자

I. 연구목적 및 필요성

대한민국 원과의 환율은 2010년 이후로 들어 1,000원부터 1,200원 사이를 행보하고 있었는데, 2021년 기준으로는 1,150원 부근에서 움직이고 있다. 2014년에 1,000원까지 떨어졌고 2015년부터 1,100원대로 올랐으며 2017년 11월 중순 부터 1,100원 이하가 되었다가 2018년 여름부터는 1,100원 초중반대를 유지하고 있다가 2019년 5월부터 1,100원 중후반대로 올랐다.(일부는 1,200원을 넘겼다.) 그러다가 2020년 3월에는 1,280원까지 올랐고 2022년 5월 이후로 1,290원을 넘었다. 2022년 6월에 1,300원대를 돌파했고 2022년 9월엔 1400원을 넘었다. 보통 사람들은 달러당 1,000원에서 1,200원 정도로 계산하는데, 2022년들어 여러 상황이 겹치면서 1400원까지 오른 실정이다. 그리고 17일 원/달러 환율은 1339.1원에 마감했다.

2022년에만 달러의 원화 환율은 1100~1400원대를 행보했다. 미국이 세계적 강국인 만큼(2022년 범세계적 경기 침체에도 3분기 플러스 성장률을 보임) 현금을 달러로 보유하거나, 미국 주식을 통해 제테크를 하는 사람들이 많다. 이들에게는 이렇게 큰 달러의 변동에 혼란스러울 것이다. 이에 달러 가격을 예측하는 것은 중요할 것이다.

II. 배경지식

1. 환율의 정의

환율(exchange rate)이란 외국 통화 한 단위를 받기 위해 자국 통화를 몇 단위 지불해야 하는가를 나타내는 것으로 자국 통화와 외국 통화간의 교환비율을 의미하며 두 나라 통화의 상대적 가치를 말한다.

2. 환율의 변동요인

환율을 결정하는 가장 근본적인 요인으로는 해당국가와 상대국의 물가수준 변동을 들 수 있다. 통화가치는 재화, 서비스, 자본 등에 대한 구매력의 척도이므로 결국 환율은 상대 물가수준으로 가늠되는 상대적 구매력에 의해 결정되기 때문이다.

장기적으로 환율에 영향을 미치는 또 다른 요소로 생산성의 변화를 들 수 있다. 예를 들어 한나라의 생산성이 다른 나라보다 더 빠른 속도로 향상(악화)될 경우 자국통화는 절상(절하)된다. 이는 생산성이 개선될 경우 재화생산에 필요한 비용이 절감되어 더 싼 값에 재화를 공급할 수 있게 되어 물가가 하락하고 통화가치는 올라가게 된다.

중기적 관점에서 보면 환율에 영향을 미치는 요인으로 대외거래, 거시경제정책 등을 들 수 있다.

대외거래 결과 국제수지가 흑자를 보이면 외환의 공급이 늘어나므로 환율은 하락하고, 국제수지가 적자를 보여 외환의 초과수요가 지속되면 환율은 상승하게 된다. 통화정책 등 거시경제정책도 환율에 영향을 미친다. 통화정책을 긴축적으로 운용하면 통화공급이 감소하여 외국의 통화량에 변화가 없다면 원화의 상대적인 공급이 줄어들어 환율이 하락(원화절상)한다.

단기적으로 환율은 외환시장 참가자들의 기대나 주변국의 환율 변동, 각종 뉴스 등에 따라 영향을 받는다.

IV. 연구내용 및 방법

유사패턴을 통해 데이터를 분석하고 향후의 데이터의 흐름을 예측한다. 쉽게 말해 데이터의 패턴을 찾아 앞으로의 방향성을 예측하는 것이다.

- 1. N일의 데이터를 추출하고 과거의 데이터와 비교하여 달러의 가격을 예측하는 모형을 만든다.
- 2. (N, P) N을 추출할 데이터, P를 예측기간이라고 할 때, (30, 15), (90, 45), (150, 75), (300, 150)간격으로 달러의 가격을 예측해본다.

<데이터 분석을 위한 라이브러리>

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib as mpl
import matplotlib.pyplot as plt
import FinanceDataReader as fdr

from numpy import dot
from numpy.linalg import norm
import numpy as np

mpl.rc('font', family='Malgun Gothic')
```

<달러 원화 환율에 대한 데이터를 변수에 저장한다.>

```
# finance datareader로부터 데이터 불러오기
data = fdr.DataReader('USD/KRW')
data
```

	Open	High	Low	Close	Adj Close	Volume
Date						
2003-12-01	1197.300049	1203.500000	1195.000000	1198.400024	1198.400024	0.0
2003-12-02	1198.099976	1198.500000	1186.199951	1195.000000	1195.000000	0.0
2003-12-03	1195.000000	1198.400024	1192.500000	1192.500000	1192.500000	0.0
2003-12-04	1192.599976	1197.900024	1174.800049	1190.099976	1190.099976	0.0
2003-12-05	1189.400024	1193.800049	1178.599976	1184.699951	1184.699951	0.0
...
2022-11-15	1324.079956	1326.729980	1306.060059	1324.079956	1324.079956	0.0
2022-11-16	1316.260010	1333.810059	1314.420044	1316.260010	1316.260010	0.0
2022-11-17	1336.430054	1351.140015	1329.349976	1336.430054	1336.430054	0.0
2022-11-18	1342.750000	1343.359985	1334.520020	1342.750000	1342.750000	0.0
2022-11-21	1339.689941	1359.030029	1339.689941	1357.630005	1357.630005	0.0

4951 rows x 6 columns

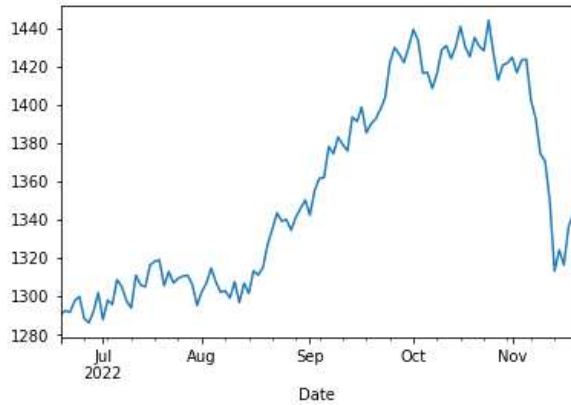
```
y=data['Close']
close=y.fillna(method='ffill')
```

close	
Date	
2003-12-01	1198.400024
2003-12-02	1195.000000
2003-12-03	1192.500000
2003-12-04	1190.099976
2003-12-05	1184.699951
...	
2022-11-15	1324.079956
2022-11-16	1316.260010
2022-11-17	1336.430054
2022-11-18	1342.750000
2022-11-21	1357.630005
Name: Close, Length: 4951, dtype: float64	

<비교구간을 설정한다.>

```
# 비교 기준 구간  
start_date = '2022-06-19'  
end_date = '2022-11-19'
```

```
# 기준 구간 시계열 차트  
close[start_date:end_date].plot();
```



<비교구간의 데이터를 정규화를 통해 0과 1 사이의 값으로 변환시킨다.>

```
base = close[start_date:end_date]
```

```
Date  
2022-06-20    1290.500000  
2022-06-21    1292.380005  
2022-06-22    1291.599976  
2022-06-23    1297.680054  
2022-06-24    1299.810059  
...  
2022-11-14    1313.030029  
2022-11-15    1324.079956  
2022-11-16    1316.260010  
2022-11-17    1336.430054  
2022-11-18    1342.750000  
Name: Close, Length: 110, dtype: float64
```

```
base_norm = (base - base.min()) / (base.max() - base.min())  
base_norm
```

```
Date  
2022-06-20    0.027257  
2022-06-21    0.039174  
2022-06-22    0.034229  
2022-06-23    0.072769  
2022-06-24    0.086271  
...  
2022-11-14    0.170069  
2022-11-15    0.240112  
2022-11-16    0.190543  
2022-11-17    0.318396  
2022-11-18    0.358456  
Name: Close, Length: 110, dtype: float64
```

<코사인 유사도의 함수를 정의한다.>

```
: from numpy import dot
  from numpy.linalg import norm
  import numpy as np

  # 코사인 유사도 함수
  def cosine(A, B):
      return dot(A, B)/(norm(A)*norm(B))
```

<비교구간의 크기, 예측 기간, 검색 횟수를 다음의 변수로 지정한다.>

```
# 윈도우 사이즈
window_size = len(base)

# 예측 기간
next_date = 200

# 검색 횟수
moving_cnt = len(close)-window_size +1
```

<다음의 모형을 이용해 비교구간과 데이터의 코사인 유사도를 비교한다.>

비교구간과 데이터의 n번째 값에서부터 윈도우 사이즈만큼을 코사인 유사도를 통해 sim_list의 변수에 나열한다.
그리고 sim_list를 오름차순으로 표현한다.

```
: # 유사도 저장 딕셔너리
sim_list = []

for i in range(moving_cnt):
    target = close[i:i+window_size]

    # Normalize
    target_norm = (target - target.min()) / (target.max() - target.min())

    # 코사인 유사도 저장
    cos_similarity = cosine(base_norm, target_norm)

    # 코사인 유사도 <- i(인덱스), 시계열데이터 함께 저장
    sim_list.append(cos_similarity)
```

```
pd.Series(sim_list).sort_values(ascending=False).head(5)
```

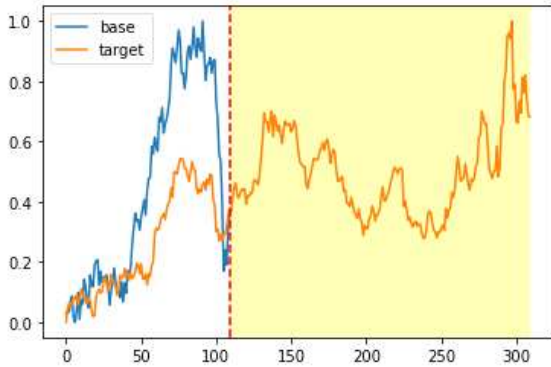
```
4840    1.000000
4839    0.993992
4841    0.993894
4838    0.988574
4837    0.983165
4836    0.977429
dtype: float64
```

<코사인 유사도가 1가 가까운 패턴을 통해 예측기간만큼 데이터를 예측한다.>

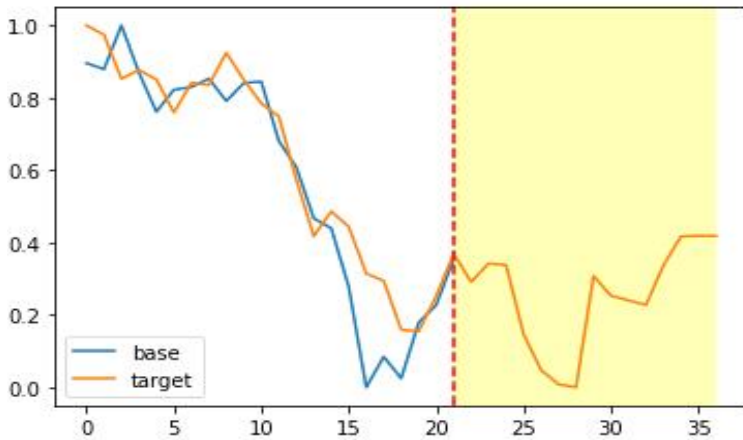
```
idx = 3959

top_ = close[idx:idx+window_size+next_date]
top_norm = (top_ - top_.min()) / (top_.max() - top_.min())

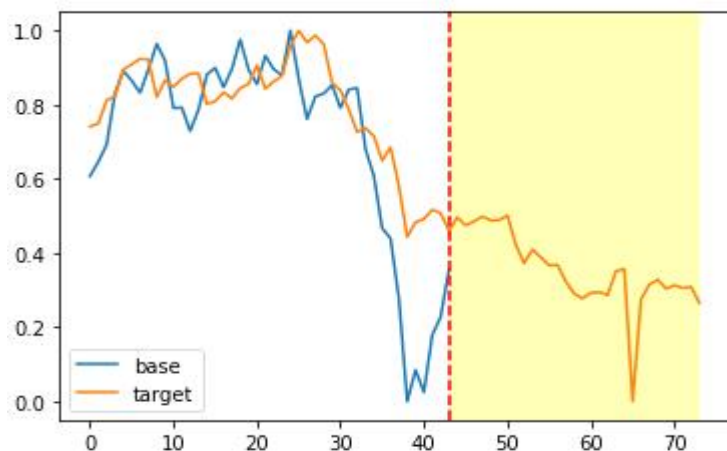
plt.plot(base_norm.values, label='base')
plt.plot(top_norm.values, label='target')
plt.axvline(x=len(base_norm)-1, c='r', linestyle='--')
plt.axvspan(len(base_norm.values)-1, len(top_norm.values)-1, facecolor='yellow', alpha=0.3)
plt.legend()
plt.show()
```



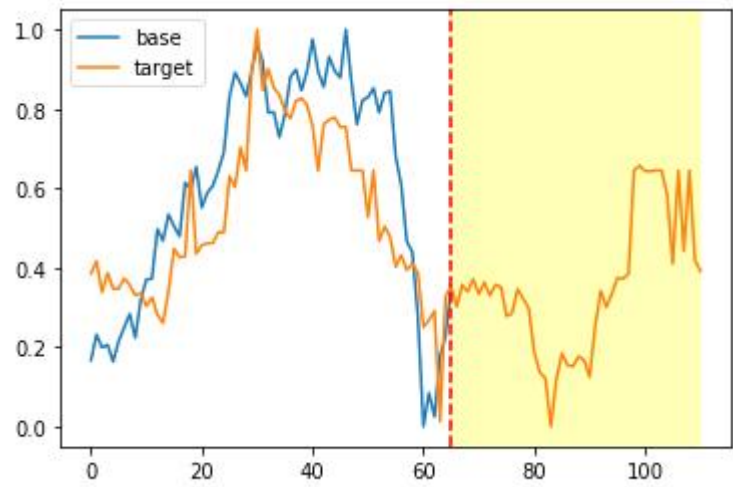
(30, 15)



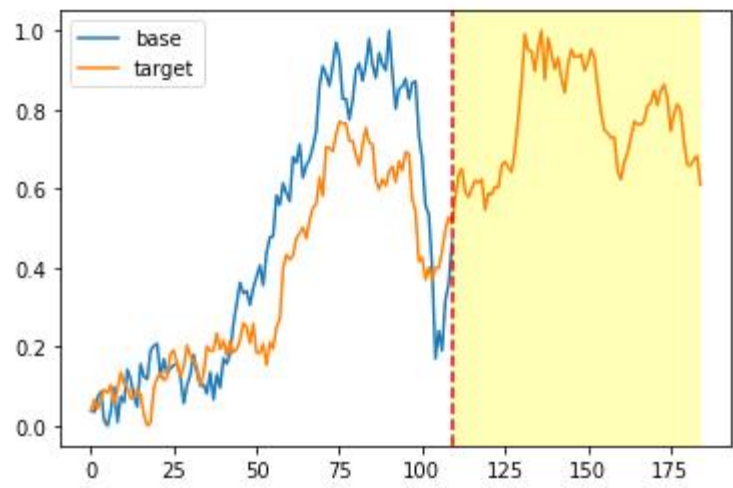
(60, 30)



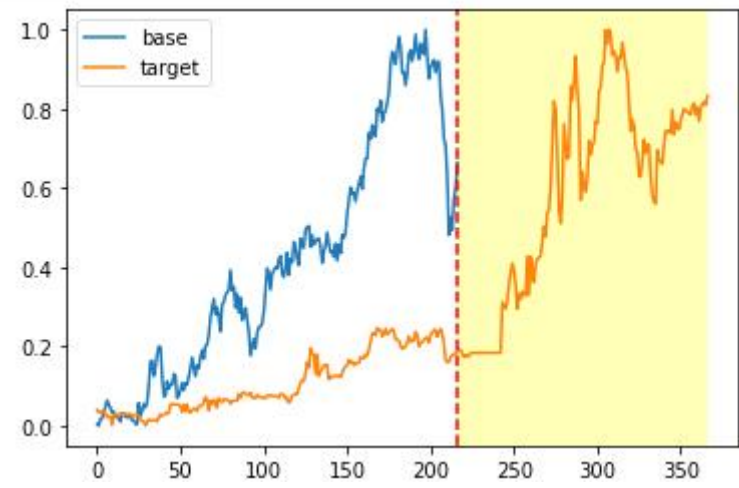
(90, 45)



(150, 75)



(300, 150)



결론
최근 달러의 가격 1400을 찍고 다시 1300원대로 내려왔다. 하지만 과거의 달러 가격을 미루어 보았을 때 다시 1400원대에 진입할 가능성이 높다.