

CP1_Project AI12_2team

리셀플랫폼을 활용한 마케팅 전략



adidas®



AI_12_김나연
AI_12_송나현
AI_12_안원석
AI_12_황건하



Contents

01. 프로젝트 개요

데이터 소개 및 분석 배경과 목적

02. 프로젝트 팀 구성 및 역할

팀 구성 및 역할, 협업 방식 소개

03. 프로젝트 수행 절차 및 방법

프로젝트 수행 절차, 방법 소개

04. 프로젝트 결과 및 평가

결과 평가, 향후 보완점

프로젝트 개요



리셀 플랫폼이란?



리셀 플랫폼은 한정판, 명품 제품의 입찰가를 실시간으로 공개한다. 이용자들은 원하는 제품의 시세를 확인하고 거래를 할 수 있다.

리셀 플랫폼의 거래 방법은 '즉시 거래'와 '입찰' 2가지다. 즉시 거래는 별도의 입찰 과정을 생략한다. 구매자가 즉시 구매를 원할 경우 판매자가 제시한 제품 가격을 선택하면 바로 구매할 수 있다. 반대로 판매자는 즉시 판매를 원하면 구매자가 제시한 가격을 선택하면 거래가 바로 이뤄진다.

플랫폼은 상품 상태와 정품 여부를 검수하고, 통과한 상품은 구매회원에게 배송하는 식으로 중개 역할을 하여 고객들이 안심하고 이용할 수 있게 한다.

프로젝트 개요

- 프로젝트 주제 선정 배경

최근 리셀이라는 소비패턴에 재테크가 더해져 ‘리셀테크’가 등장하였다. ‘리셀테크’는 ‘리셀(resell)’과 ‘재테크’를 합친 말로, 소위 명품이나 소장 가치가 있는 제품을 구매한 후 나중에 좀 더 비싼 가격에 제품을 되팔아 이익을 보는 재테크 방식을 말한다. 그중에서도 가장 활성화된 부문은 스니커즈 리셀 시장이다. 스니커즈 리셀이 고가의 명품 대비 비교적 낮은 금액대로 진입할 수 있어 MZ세대의 대표 재테크 방법으로 급부상하고 있다.

- 프로젝트 목적 및 기대효과

재밌게 해 볼 수 있는 ‘신발 제테크’를 통해 리셀 플랫폼 인식 확대하고 고객들이 보유한 신발의 잠재 가치를 쉽게 알 수 있게하여 신규 고객을 확보 및 다방면으로의 리셀 시장 활성화를 목표로 한다. 또한, 트렌드 분석을 통해 리셀 플랫폼 고객의 니즈(수요)를 파악하여 적합한 마케팅 전략을 수립하고자 한다.

최근 리셀 가격이 치솟으며 ‘재테크 효자’가 된 스니커즈 상품들

조던1x 오프화이트 레트로하이 시카고 더텐		발매가 22만3500원 ▼ 최근 거래가 1100만 원
조던1x 오프화이트 레트로 하이 화이트		22만3500원 ▼ 650만 원
나이키x 트래비스 스콧 SB 덩크 로		17만9000원 ▼ 270만 원
조던1x 트래비스 스콧 레트로 하이 OG SP 모카		23만9000원 ▼ 224만9000원
나이키x 피스마이너스원 에어포스 1 로 파라노이즈 2.0		21만9000원 ▼ 73만6000원
조던1x트래비스 스콧 프래그먼트 레트로 로 OG SP 밀리터리 블루		18만9000원 ▼ 140만 원
나이키x피스마이너스원 원도 1 화이트 크림 홈페이지 캡처		21만9000원 ▼ 51만9000원



프로젝트 개요

- 프로젝트 구조





프로젝트 개요

- 개발 환경

1. 소프트 웨어

Bootstrap : 웹 사이트나 응용 프로그램을 작성하기 위해 사용하는 무료 소프트웨어 도구 모음

VSCode : 에디터

Google Colab: 구글에서 제공하는 jupyter notebook

2. 라이브러리

scikit-learn: 데이터 분석 및 머신러닝 적용을 위한 파이썬 기반 라이브러리

selenium: 동적 웹 크롤링

Matplotlib, Seaborn, chart.js: 시각화 라이브러리

3. 프레임워크

Flask: 파이썬으로 작성된 마이크로 웹 프레임워크



프로젝트 팀 구성 및 역할

- 팀원 구성 및 역할

AI12_김나연: 데이터 수집, 데이터 전처리/EDA 및 모델링, ppt 제작, 발표

AI12_송나현: 데이터 수집, 데이터 전처리/분석 및 모델링, ppt 제작, 발표

AI12_안원석: 데이터 수집, 웹 서비스/ 대시보드 구현 및 배포, 발표

AI12_황건하: 데이터 수집, 데이터 전처리/EDA 및 모델링, ppt 제작, 발표

- team2 협업 방식

실시간 디스코드를 활용한 자료 및 의견 공유, 게더타운에서 팀원과 코치님 의견 공유



프로젝트 수행 절차 및 방법

구분	기간	활동	비고
사전 기획	6/24(금) ~ 6/28(화)	- 프로젝트 기획 및 주제선정 - 기획서 작성	- 아이디어 수집
데이터 수집	6/28(화) ~ 6/30(목)	- 데이터 크롤링 - 필요한 데이터 추가 수집	- 팀별 회의
데이터 전처리	6/30(목) ~ 7/3(일)	- 데이터 정제 및 EDA	- 팀별 회의
모델링	7/2(토) ~ 7/5(화)	- 베이스라인 모델 제작 - 모델 구현 및 보완	- 팀별 회의 - 발표자료 제작
서비스 구축	7/2(토) ~ 7/6(수)	- 웹서비스 구현 - 대시보드 제작	- 프로젝트 마무리
총 개발기간	6/24(금) ~ 7/6(수) (약 2주)	-	-



프로젝트 결과 및 평가



프로젝트 개요

데이터 소개

- train data

kaggle, StockX 2019 Data Contest에서 제공한 데이터,
10000개의 신발 데이터 (99956 rows × 8 columns)

- test data

kaggle, StockX 최근 출시된 신발 데이터 (1000 rows × 17 columns)

- 트렌드 분석에 사용한 데이터

Kream 사이트 크롤링



프로젝트 결과 및 평가

train data features

- Order Date	:	주문일
- Brand	:	브랜드 이름
- Sneaker Name	:	신발 상품명
- Sale Price	:	판매가
- Retail Price	:	소매가
- Release Date	:	출시일
- Shoe Size	:	신발 사이즈
- Buyer Region	:	구매자 지역



프로젝트 결과 및 평가

time gap 칼럼 생성 (Order Date(주문일) - Release Date(출시일))

time gap
342
282
282

- 출시일로 부터 시간이 지날 수록 판매가격이 낮아질거라고 추정하여, 주문일에서 출시일을 빼주는 time gap column 생성



프로젝트 결과 및 평가

Brand Adidas, Jordan, Nike 칼럼 생성

- 상품명에서 브랜드 이름을 추출하여 숫자형 데이터로 칼럼 생성

Sneaker Name
Adidas Yeezy Boost 350 Low V2 Beluga
Adidas Yeezy Boost 350 V2 Core Black Copper



Brand_Adidas	Brand_Jordan	Brand_Nike
1.0	0.0	0.0
1.0	0.0	0.0



Sneaker Name 비슷한 상품명 통일

- 비슷한 상품명들 사이즈, 색상 차이로 가격이 비슷한 경우가 많아 이름을 하나로 통일

Sneaker Name
Adidas Yeezy Boost 350 Low V2 Beluga
Adidas Yeezy Boost 350 V2 Core Black Copper
Adidas Yeezy Boost 350 V2 Core Black Green
Adidas Yeezy Boost 350 V2 Core Black Red
Adidas Yeezy Boost 350 V2 Core Black Red 2017



Sneaker Name
Adidas Yeezy Boost
Adidas Yeezy Boost
Adidas Yeezy Boost
Adidas Yeezy Boost
Adidas Yeezy Boost



프로젝트 결과 및 평가

Sneaker Name 원핫인코딩

Sneaker Name
Adidas Yeezy Boost
Adidas Yeezy Boost
Adidas Yeezy Boost

- 모델링을 위해 Sneaker Name을 원핫인코딩



Sneaker_Name_Adidas Yeezy Boost	Sneaker_Name_Air Jordan 1	Sneaker_Name_Nike Air Force	Sneaker_Name_Nike Air Max	Sneaker_Name_Nike Air Presto	Sneaker_Name_Nike Air VaporMax
1.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0



Order date 그레고리력 서수로 변환

- datetime 날짜 형태인 Order date를 모델링을 위해 정수 형태로 변경

Order_date
2017-09-01



Order_date
736573



프로젝트 결과 및 평가



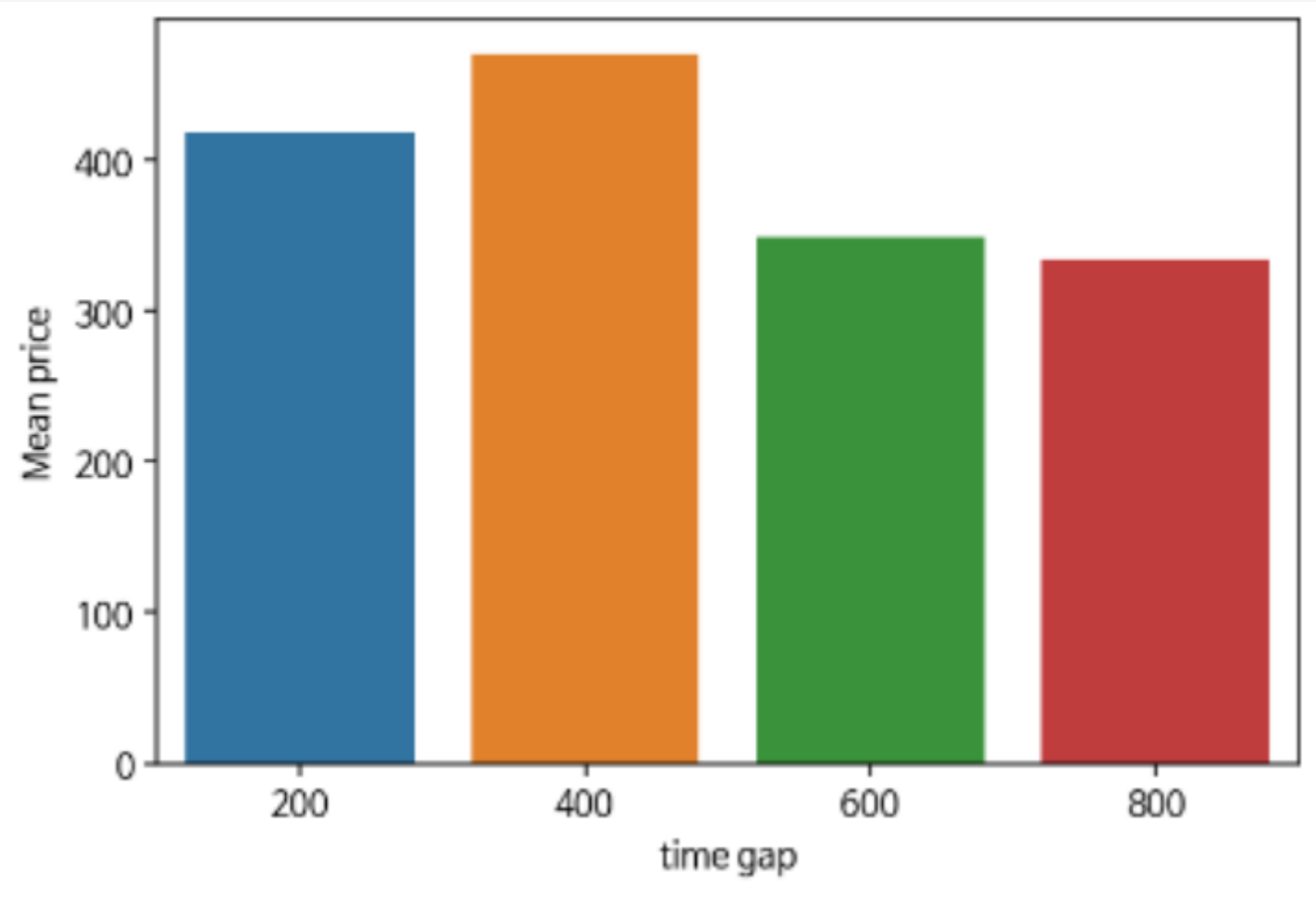
최종 데이터셋

Order_date	Retail_Price	Release_Date	time_gap	Sneaker_Name_Adidas Yeezy Boost	Sneaker_Name_Air Jordan 1	Sneaker_Name_Nike Air Force	Sneaker_Name_Nike Air Max	Sneaker_Name_Nike Air Presto
736701	220	736679	22	1.0	0.0	0.0	0.0	0.0
736880	220	736651	229	1.0	0.0	0.0	0.0	0.0

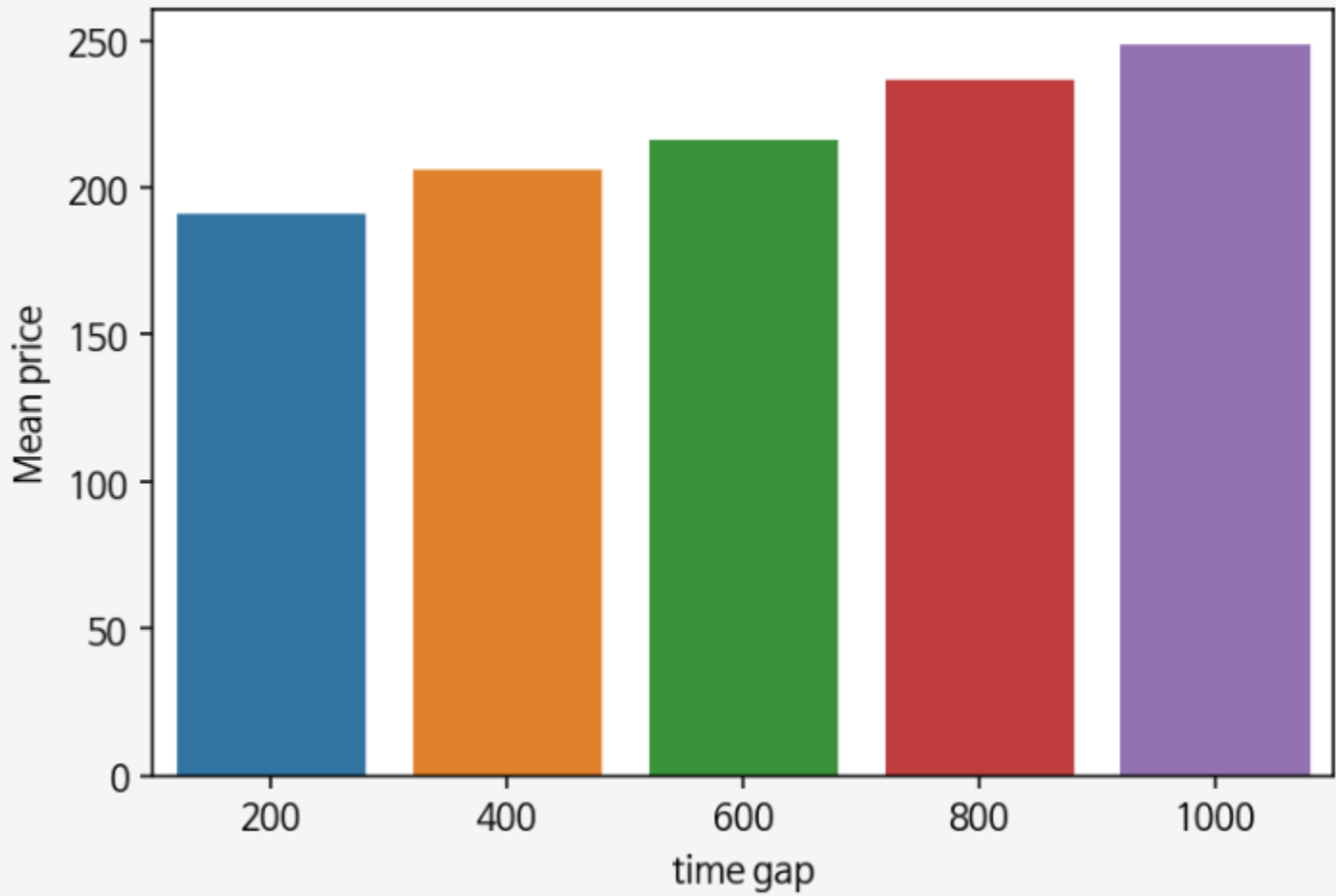
Sneaker_Name_Nike Air VaporMax	Sneaker_Name_Nike Blazer Mid	Sneaker_Name_Nike React Hyperdunk	Sneaker_Name_Nike Zoom Fly	Brand_Adidas	Brand_Jordan	Brand_Nike
0.0	0.0	0.0	0.0	1.0	0.0	0.0
0.0	0.0	0.0	0.0	1.0	0.0	0.0

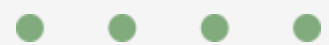
time gap 칼럼 생성 (Order Date(주문일) - Release Date(출시일))

17-19년도 Stockx



22년도 Stockx

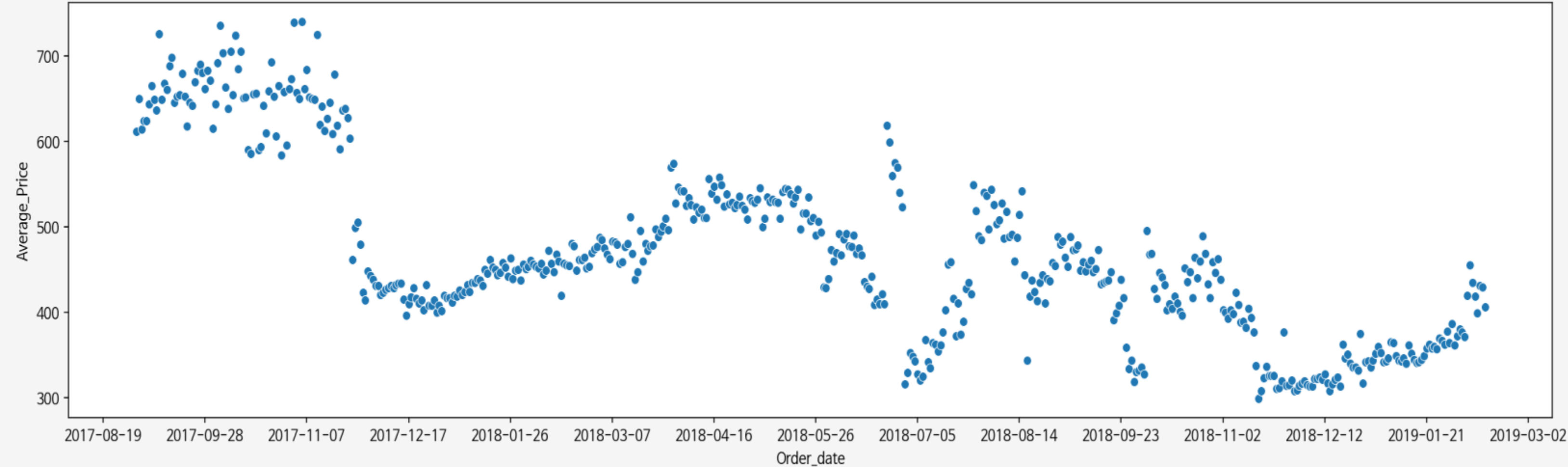


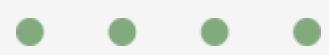


프로젝트 결과 및 평가

17-19년도 Stockx

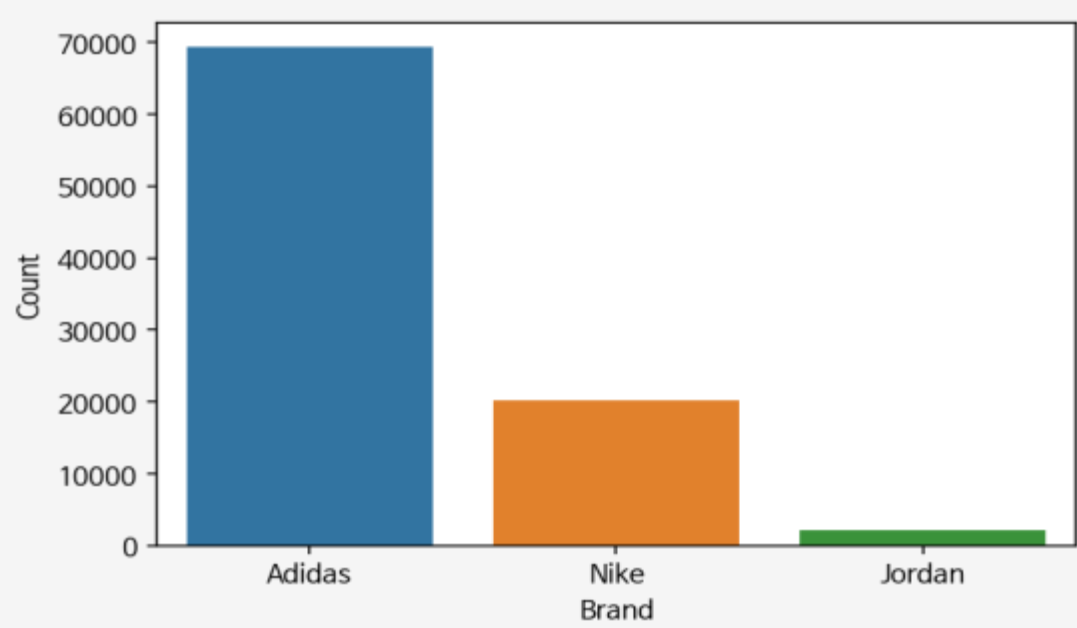
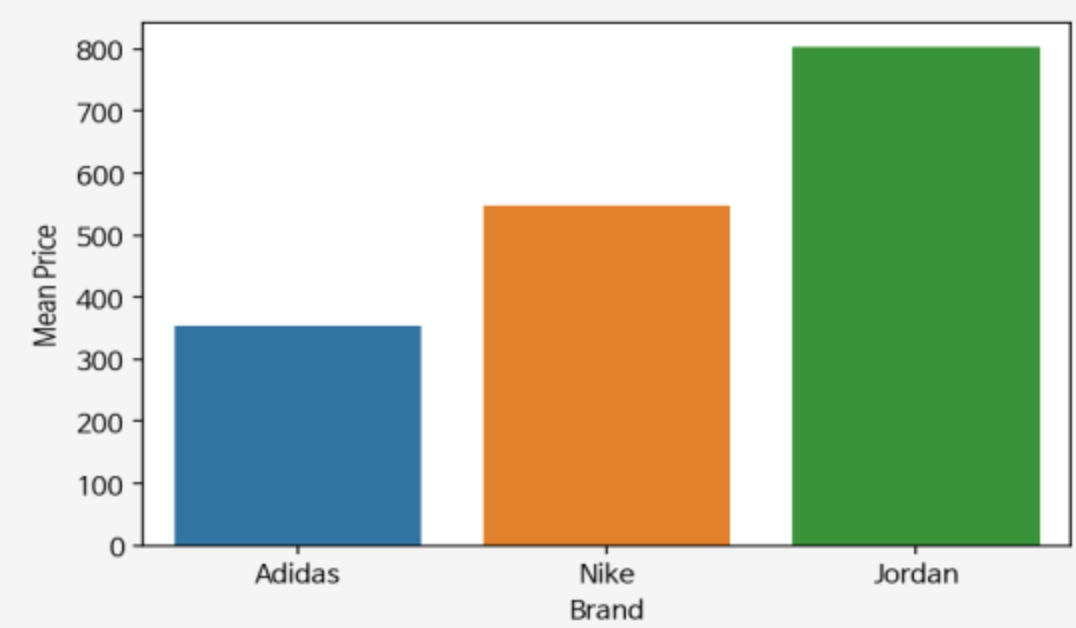
Average Daily Sale Price Over time





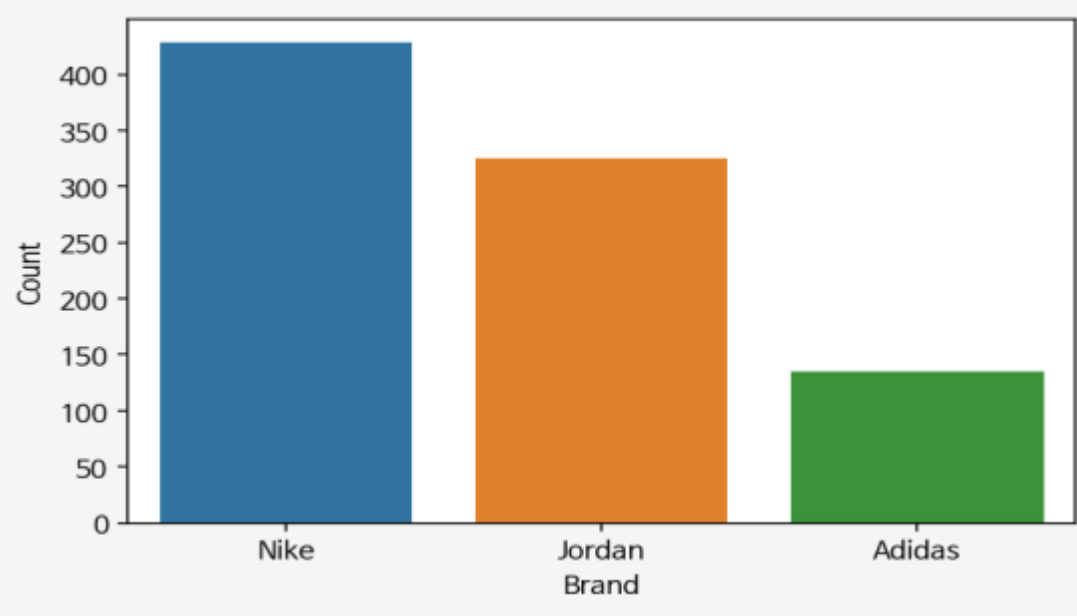
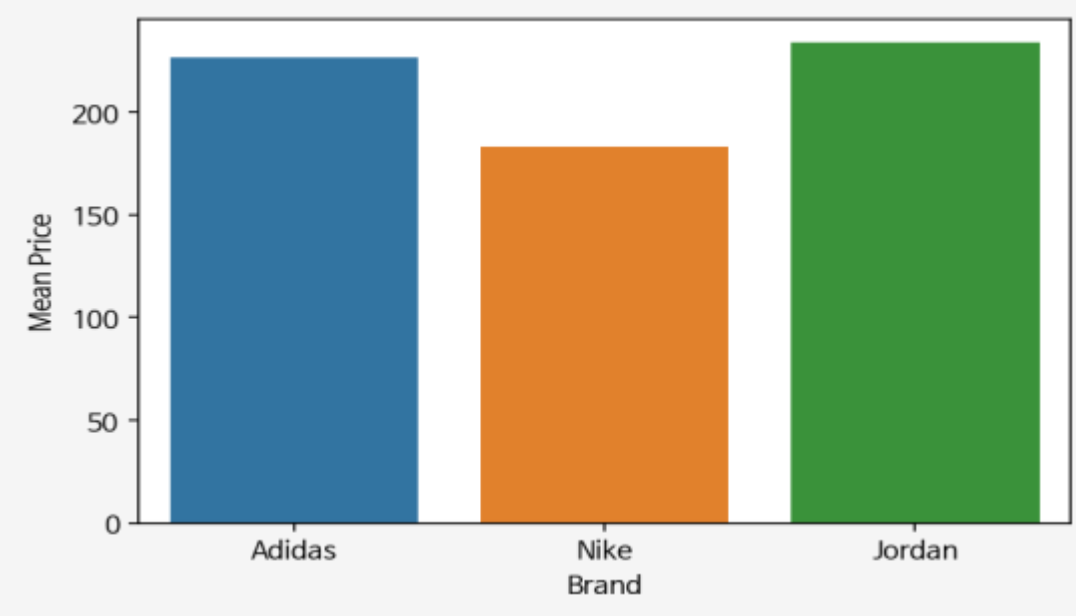
프로젝트 결과 및 평가

17-19년도 Stockx

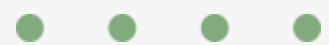


```
Adidas Yeezy Boost      69292
Nike Zoom Fly           4263
Nike Air Presto         3691
Nike Blazer Mid         3357
Nike Air VaporMax       3112
Nike Air Max            2919
Nike Air Force          2291
Air Jordan 1            2180
Nike React Hyperdunk    484
Name: Sliced Name, dtype: int64
```

22년도 Stockx



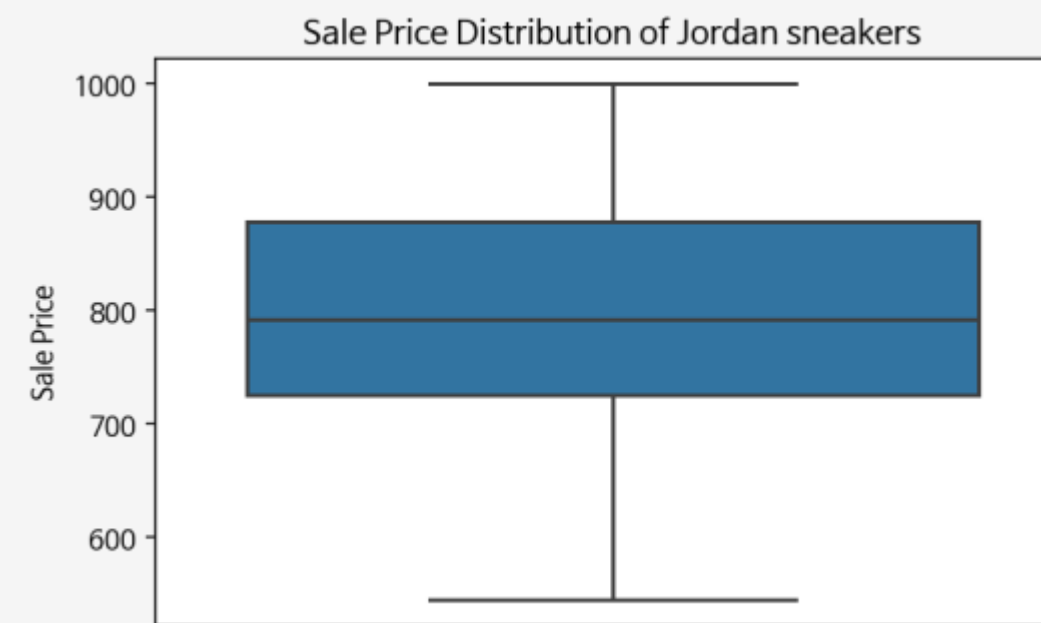
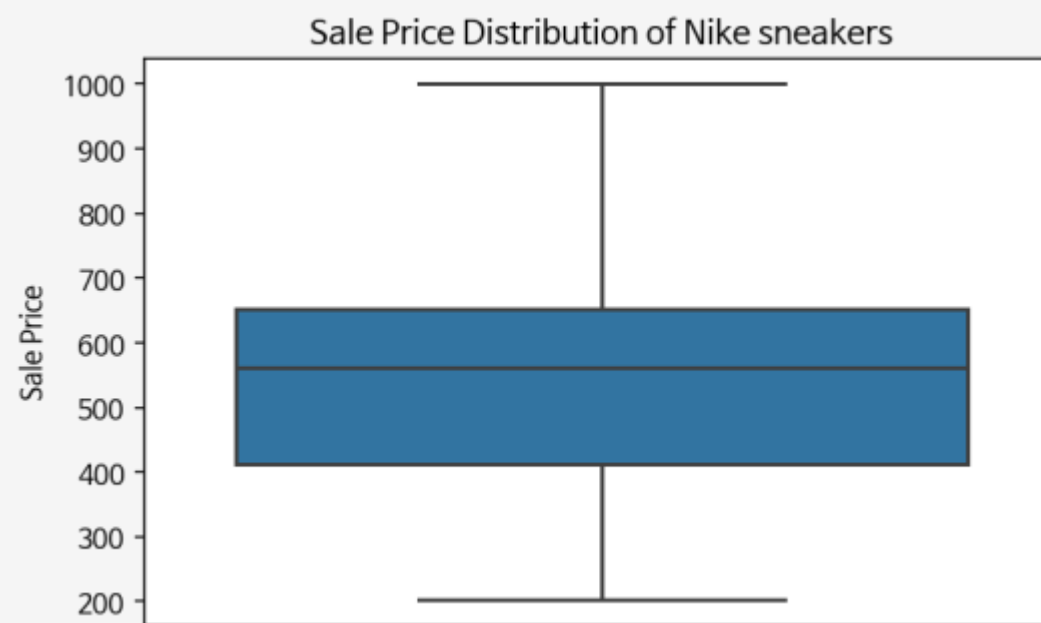
```
Nike Dunk Low           121
Jordan 1 Mid            77
Nike Air Max            66
Jordan 1 Retro          65
Nike Air Force          57
Jordan 1 Low            54
Adidas Yeezy Boost      51
Nike Dunk High          32
Nike SB Dunk            28
Jordan 4 Retro          27
New Balance 550         24
Name: Sliced Name, dtype: int64
```



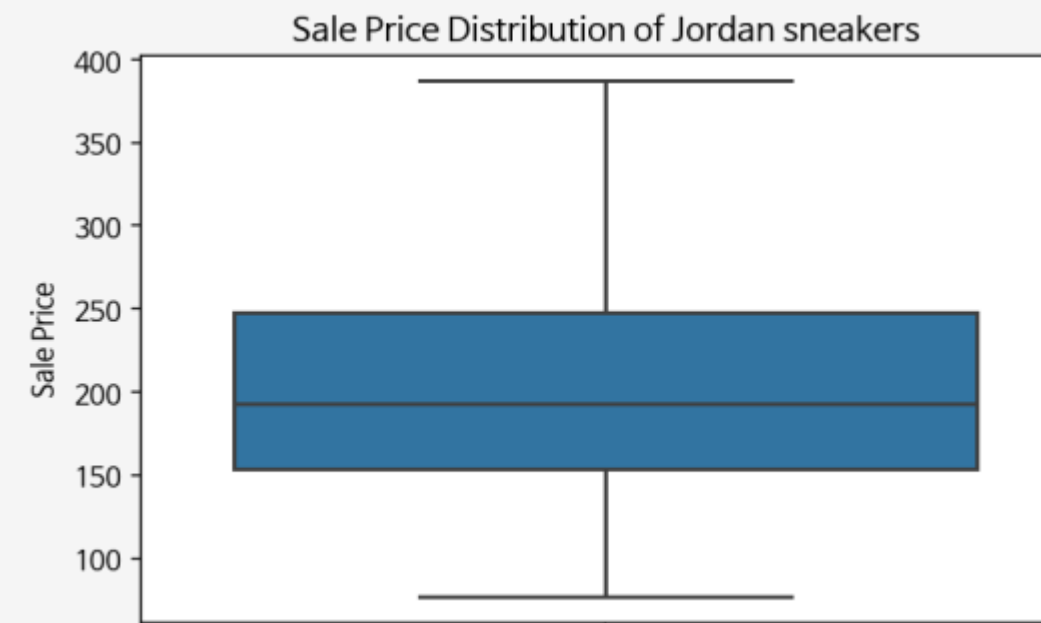
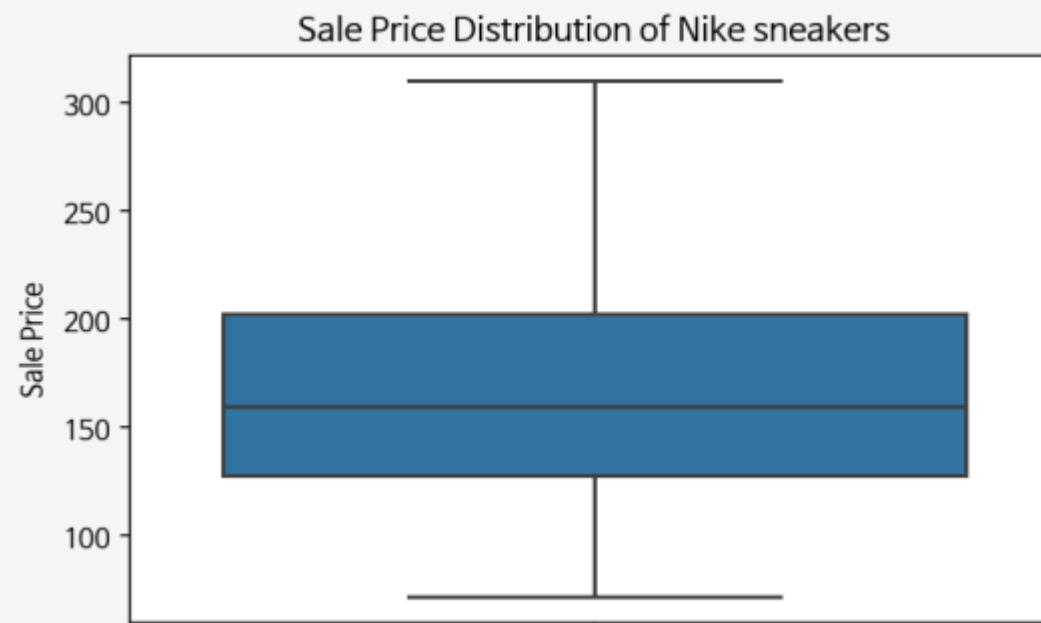
프로젝트 결과 및 평가

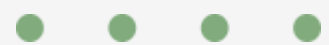


17-19년도 Stockx



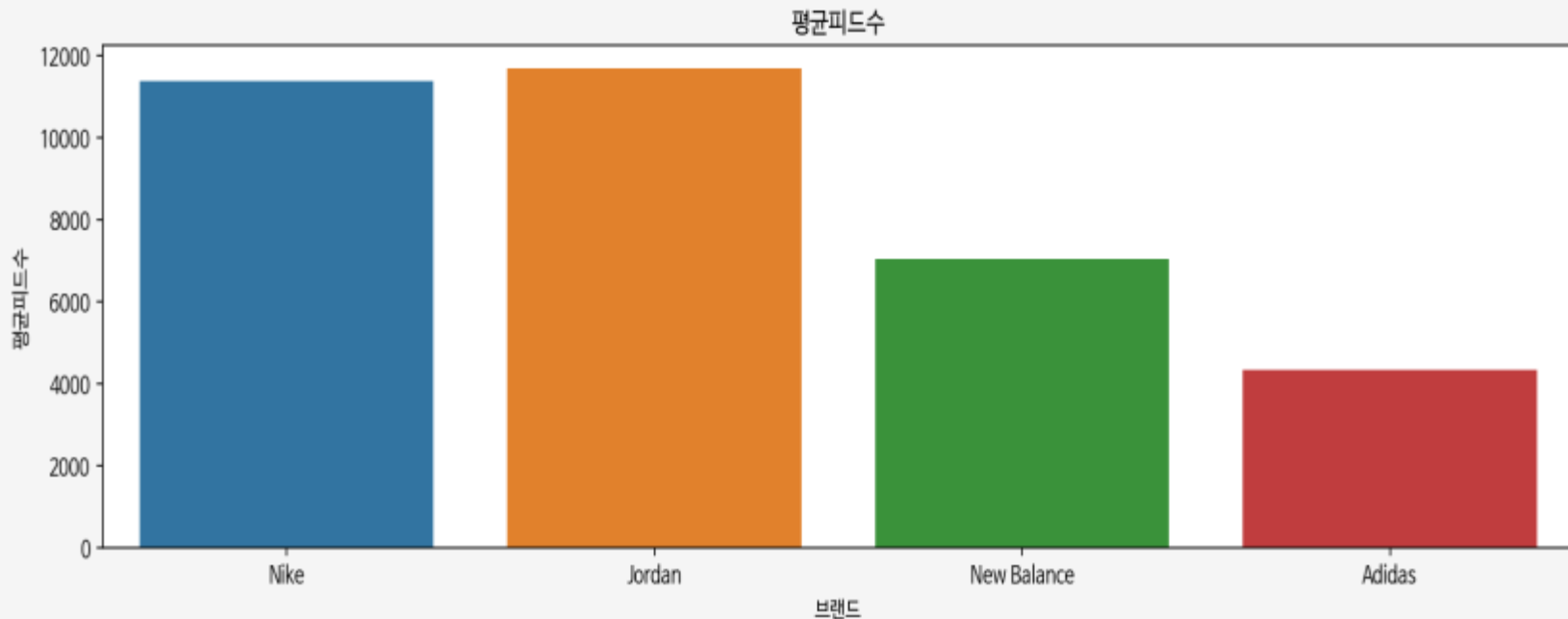
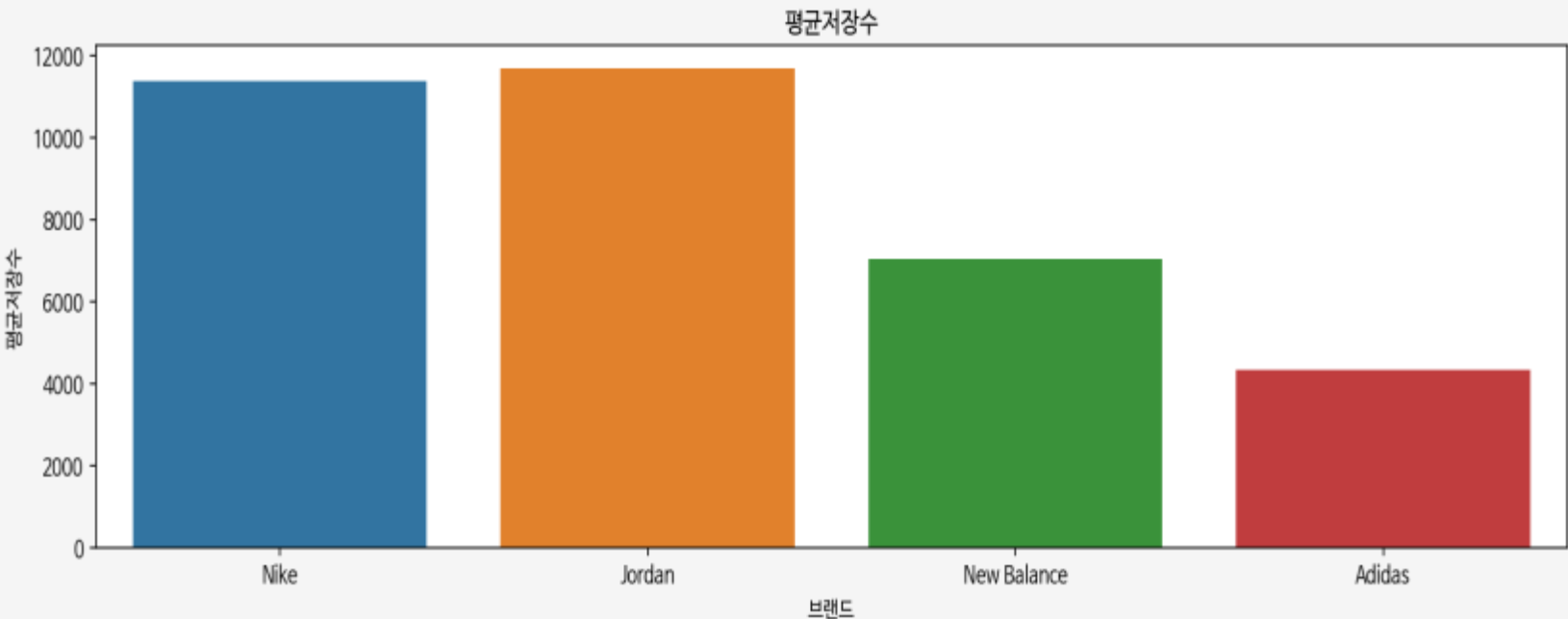
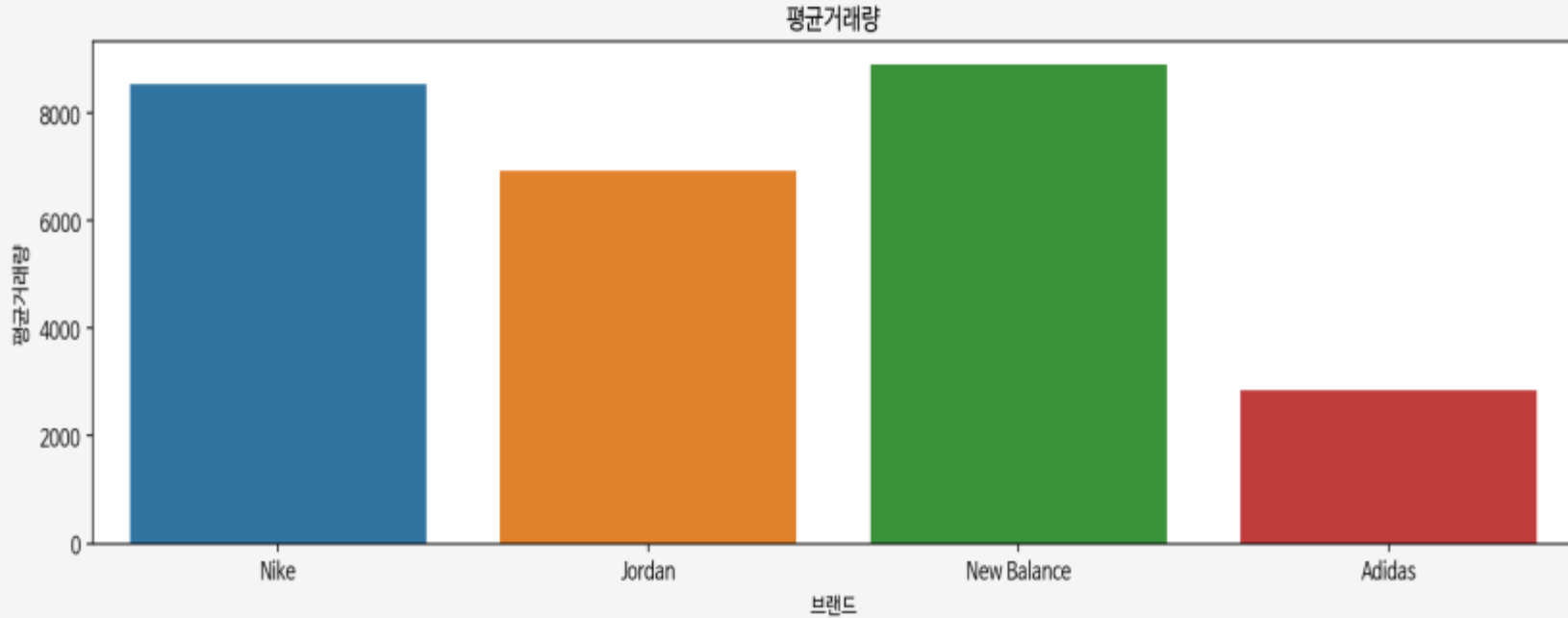
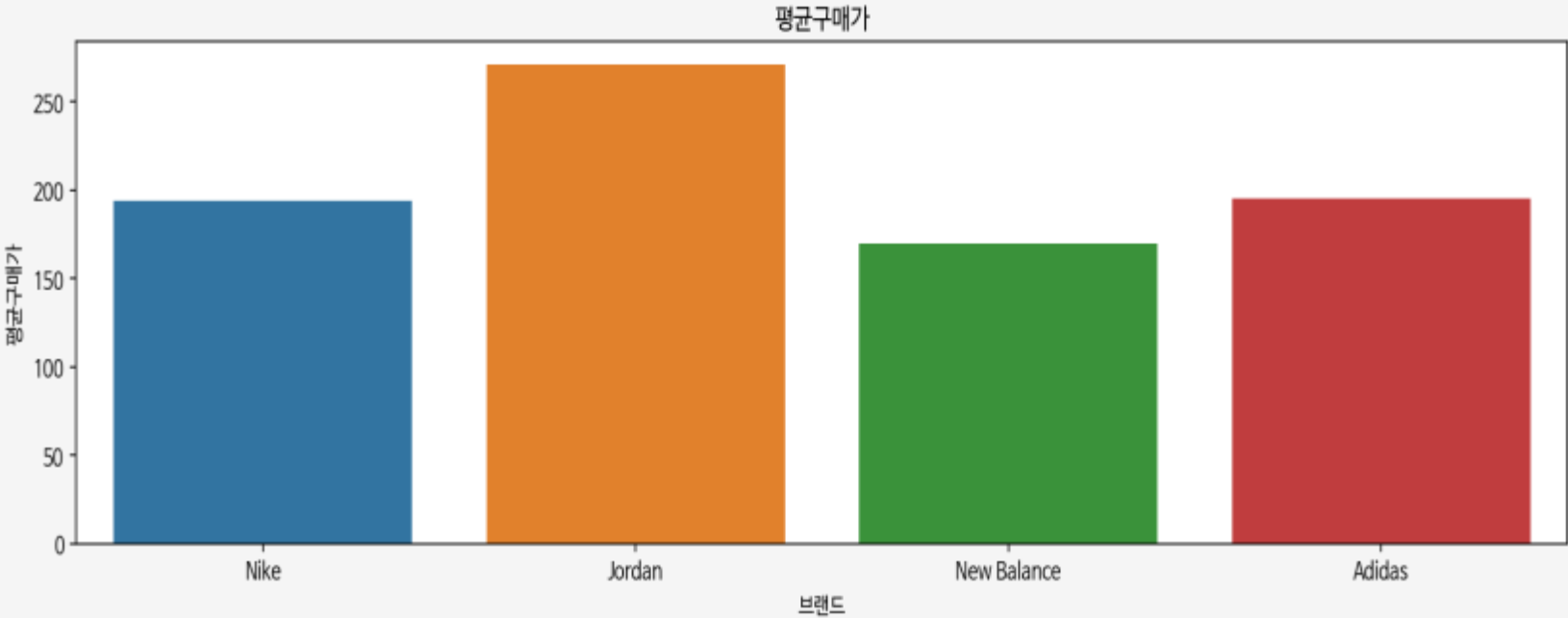
22년도 Stockx

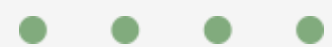




프로젝트 결과 및 평가

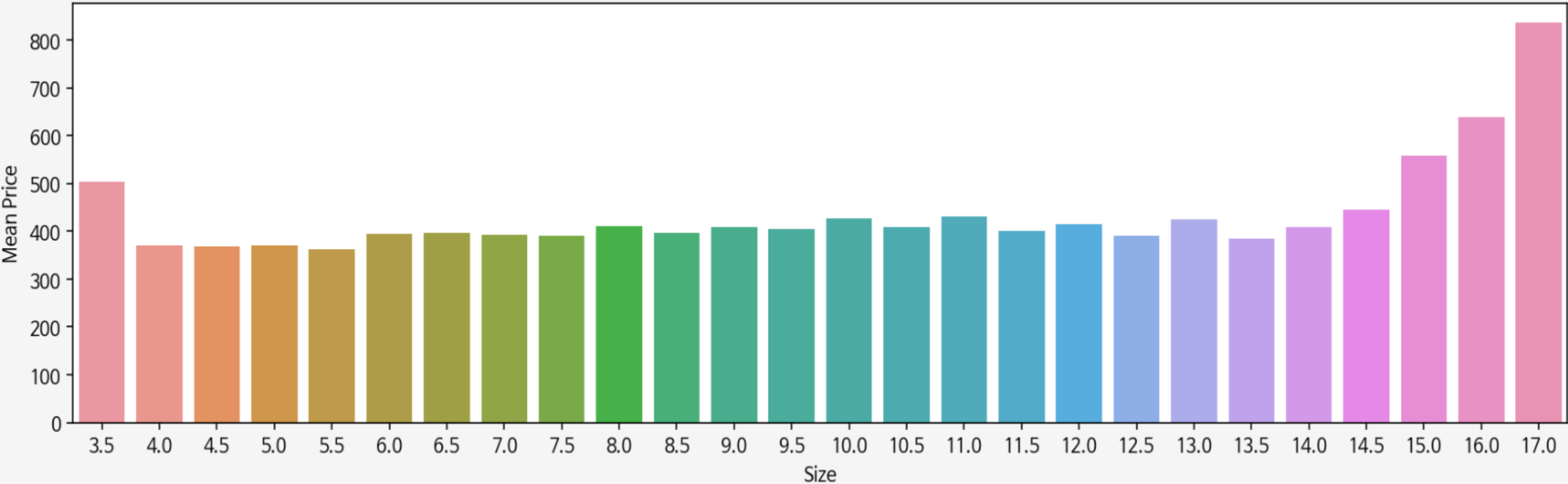
22년도 Kream

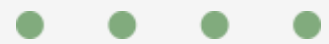




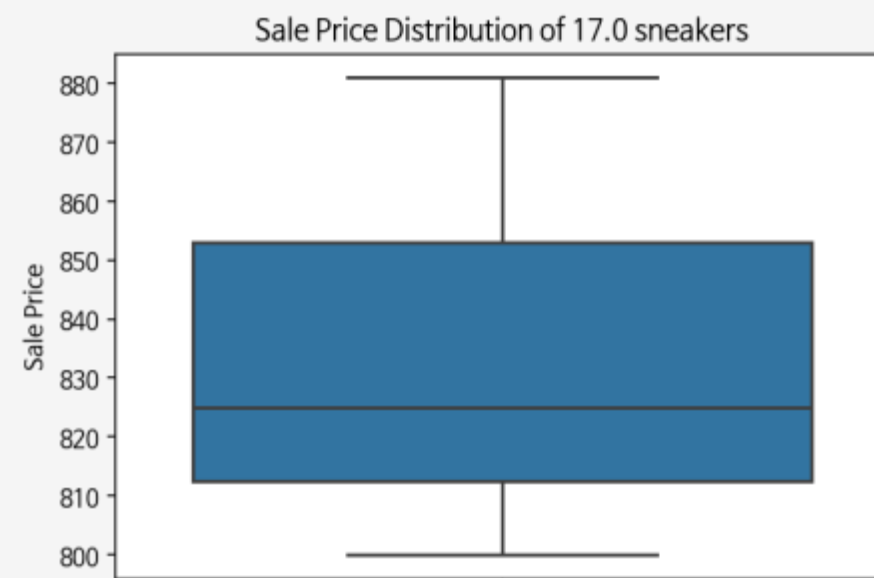
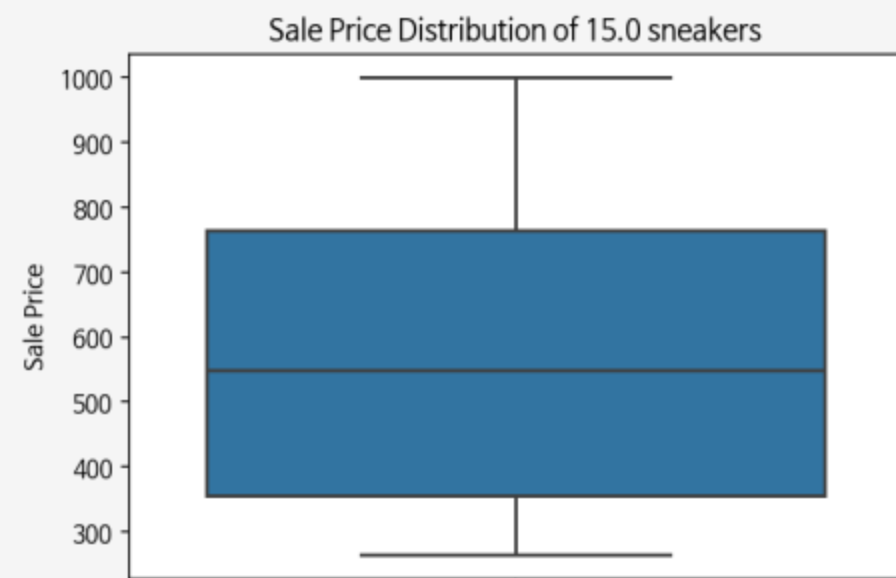
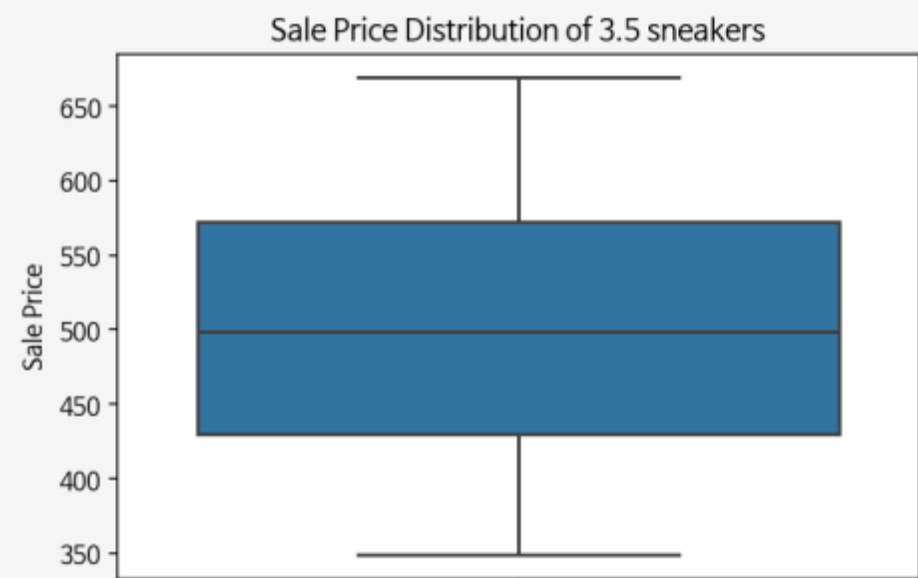
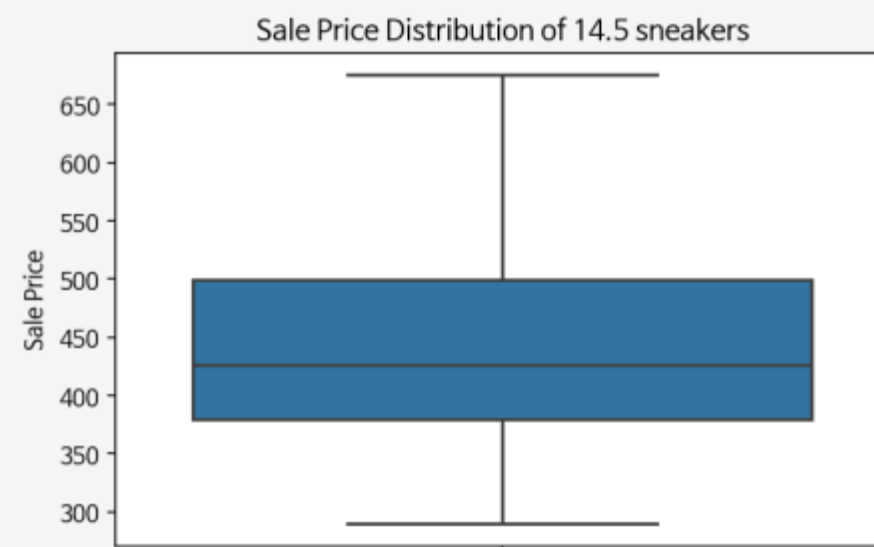
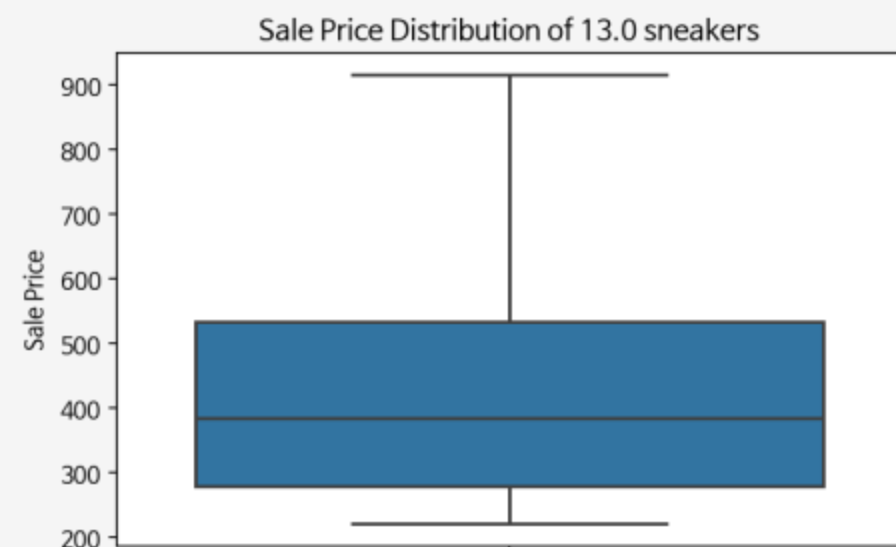
프로젝트 결과 및 평가

17-19년도 Stockx





프로젝트 결과 및 평가





프로젝트 결과 및 평가

- 모델링

RandomForest : 랜덤 포레스트는 다수의 결정 트리들을 학습하는 앙상블 방법이며
검출, 분류, 그리고 회귀 등 다양한 문제에 활용되고 있다.
랜덤 포레스트는 다양성을 극대화 하여 예측력이 우수한 편이고 다수의 트리
예측 결과를 종합하여 의사결정을 진행하기 때문에 안정성도 높다.





프로젝트 결과 및 평가



▶ Linear Regression (Baseline)

- Baseline Model로 간단한 선형회귀 적용
- 검증 데이터와 테스트 데이터를 통해 예측을 진행
- 성능 평가지표로 MAE, MSE, RMSE를 통해 확인

```
from sklearn.linear_model import LinearRegression

lm = LinearRegression()
lm.fit(OH_X_train,y_train)
```

```
LinearRegression()
```

예측값 저장, 평가 지표 적용

```
pred = lm.predict(OH_X_valid)
```

```
from sklearn import metrics
print("MAE :",metrics.mean_absolute_error(y_val,pred))
print("MSE :",metrics.mean_squared_error(y_val,pred))
print("RMSE :",np.sqrt(metrics.mean_squared_error(y_val,pred)))
```

```
MAE : 76.61400701787065
MSE : 11948.829028813856
RMSE : 109.31069951662488
```

Test 데이터로 실제 테스트

```
pred = lm.predict(OH_X_test)
```

```
from sklearn import metrics
print("MAE :",metrics.mean_absolute_error(y_test,pred))
print("MSE :",metrics.mean_squared_error(y_test,pred))
print("RMSE :",np.sqrt(metrics.mean_squared_error(y_test,pred)))
```

```
MAE : 25907203776085.824
MSE : 7.088594278305005e+26
RMSE : 26624414131216.117
```



프로젝트 결과 및 평가



▶ RandomForest

- RandomizedSearchCV 사용
- 베이스라인 모델에 비해 검증 및 테스트 데이터 모두 성능 향상 확인
- 검증 데이터 셋에서 과적합, 테스트 데이터 셋 성능이 좋지 않음

```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint, uniform
from sklearn.pipeline import make_pipeline

# pipe = make_pipeline(
#     TargetEncoder(),
#     SimpleImputer(),
#     RandomForestRegressor(random_state=2)
# )

pipe = make_pipeline(
    RandomForestRegressor(random_state=42)
)

#rf_reg = RandomForestRegressor(random_state=42)

dists = {
    'randomforestregressor__n_estimators': randint(50, 500),
    'randomforestregressor__max_depth': [5, 10, 15, None],
    'randomforestregressor__max_features': uniform(0, 1)
}

clf = RandomizedSearchCV(
    pipe,
    param_distributions=dists,
    n_iter = 10,
    cv = 3,
    scoring='neg_mean_absolute_error',
    verbose=1,
    n_jobs=-1
)

model = clf.fit(OH_X_train, y_train);
```

```
# 만들어진 모델에서 가장 성능이 좋은 모델
pipe = clf.best_estimator_

#검증, 테스트 데이터
from sklearn import metrics

y_pred = pipe.predict(OH_X_valid)
mae = metrics.mean_absolute_error(y_val, y_pred)
mse = metrics.mean_squared_error(y_val, y_pred)
rmse = np.sqrt(metrics.mean_squared_error(y_val, y_pred))
r2_score = metrics.r2_score(y_val, y_pred)
print(f'검증데이터 MAE: {mae:,.0f}')
print(f'검증데이터 MSE: {mse:,.0f}')
print(f'검증데이터 RMSE: {rmse:,.0f}')
print(f'검증데이터 R2_score: {r2_score:,.0f}')

y_pred = pipe.predict(OH_X_test)
test_mae = metrics.mean_absolute_error(y_test, y_pred)
test_mse = metrics.mean_squared_error(y_test, y_pred)
test_rmse = np.sqrt(metrics.mean_squared_error(y_test, y_pred))
test_r2_score = metrics.r2_score(y_test, y_pred)
print(f'테스트데이터 MAE: {test_mae:,.0f}')
print(f'테스트데이터 MSE: {test_mse:,.0f}')
print(f'테스트데이터 RMSE: {test_rmse:,.0f}')
print(f'테스트데이터 R2_score: {test_r2_score:,.0f}')
```

검증데이터 MAE: 27
검증데이터 MSE: 2,044
검증데이터 RMSE: 45
검증데이터 R2_score: 1
테스트데이터 MAE: 659
테스트데이터 MSE: 445,534
테스트데이터 RMSE: 667
테스트데이터 R2_score: -21



프로젝트 결과 및 평가



▶ RandomForest

- 트리 최대 깊이 조정
- RandomizedSearchCV의 n_iter 값 올려서 재시도 (10 -> 50)
- 성능에 큰 변화 없음
- 훈련 데이터셋에는 3가지의 브랜드 밖에 없어서 테스트 데이터 셋의 성능이 좋지 않은 것으로 판단

```
from sklearn.model_selection import RandomizedSearchCV
from scipy.stats import randint, uniform
from sklearn.pipeline import make_pipeline

# pipe = make_pipeline(
#     TargetEncoder(),
#     SimpleImputer(),
#     RandomForestRegressor(random_state=2)
# )

pipe = make_pipeline(
    RandomForestRegressor(random_state=42)
)
#rf_reg = RandomForestRegressor(random_state=42)

dists = {
    'randomforestregressor__n_estimators': randint(50, 500),
    'randomforestregressor__max_depth': [2, 4, 6, None],
    'randomforestregressor__max_features': uniform(0, 1)
}

clf = RandomizedSearchCV(
    pipe,
    param_distributions=dists,
    n_iter = 50,
    cv = 3,
    scoring='neg_mean_absolute_error',
    verbose=1,
    n_jobs=-1
)

model = clf.fit(OH_X_train, y_train);
```

```
# 만들어진 모델에서 가장 성능이 좋은 모델
pipe = clf.best_estimator_

#검증, 테스트 데이터
from sklearn import metrics

y_pred = pipe.predict(OH_X_valid)
mae = metrics.mean_absolute_error(y_val, y_pred)
mse = metrics.mean_squared_error(y_val, y_pred)
rmse = np.sqrt(metrics.mean_squared_error(y_val, y_pred))
r2_score = metrics.r2_score(y_val, y_pred)
print(f'검증데이터 MAE: {mae:,.2f}')
print(f'검증데이터 MSE: {mse:,.2f}')
print(f'검증데이터 RMSE: {rmse:,.2f}')
print(f'검증데이터 R2_score: {r2_score:,.2f}')

y_pred = pipe.predict(OH_X_test)
test_mae = metrics.mean_absolute_error(y_test, y_pred)
test_mse = metrics.mean_squared_error(y_test, y_pred)
test_rmse = np.sqrt(metrics.mean_squared_error(y_test, y_pred))
test_r2_score = metrics.r2_score(y_test, y_pred)
print(f'테스트데이터 MAE: {test_mae:,.2f}')
print(f'테스트데이터 MSE: {test_mse:,.2f}')
print(f'테스트데이터 RMSE: {test_rmse:,.2f}')
print(f'테스트데이터 R2_score: {test_r2_score:,.2f}')

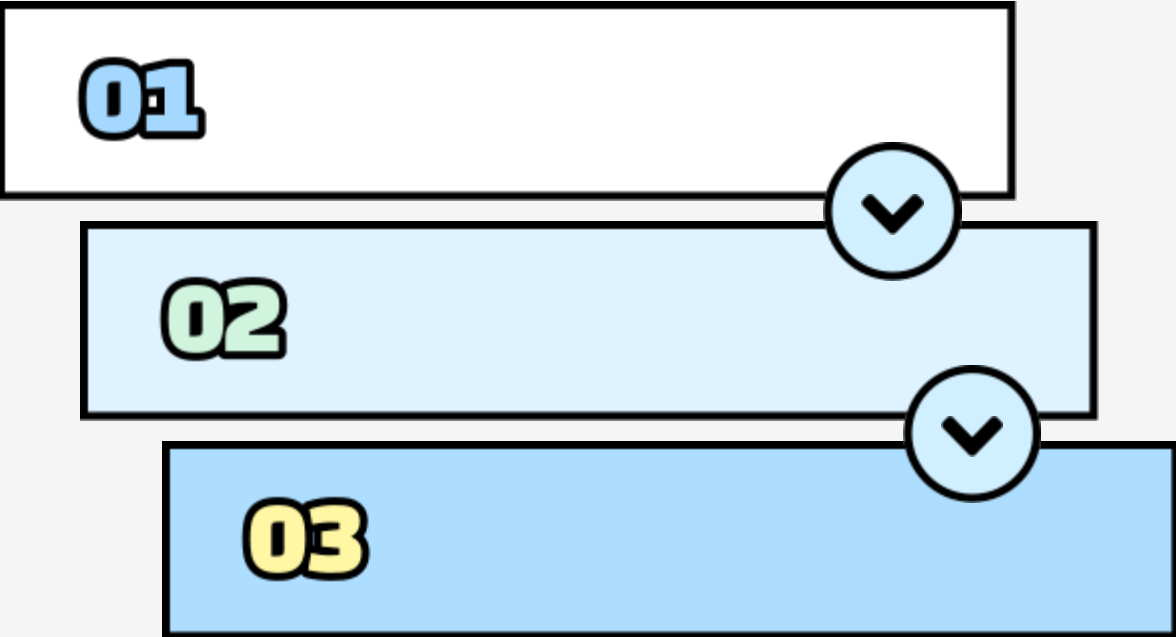
검증데이터 MAE: 26.87
검증데이터 MSE: 1,926.87
검증데이터 RMSE: 43.90
검증데이터 R2_score: 0.93
테스트데이터 MAE: 660.33
테스트데이터 MSE: 445,853.68
테스트데이터 RMSE: 667.72
테스트데이터 R2_score: -20.64
```



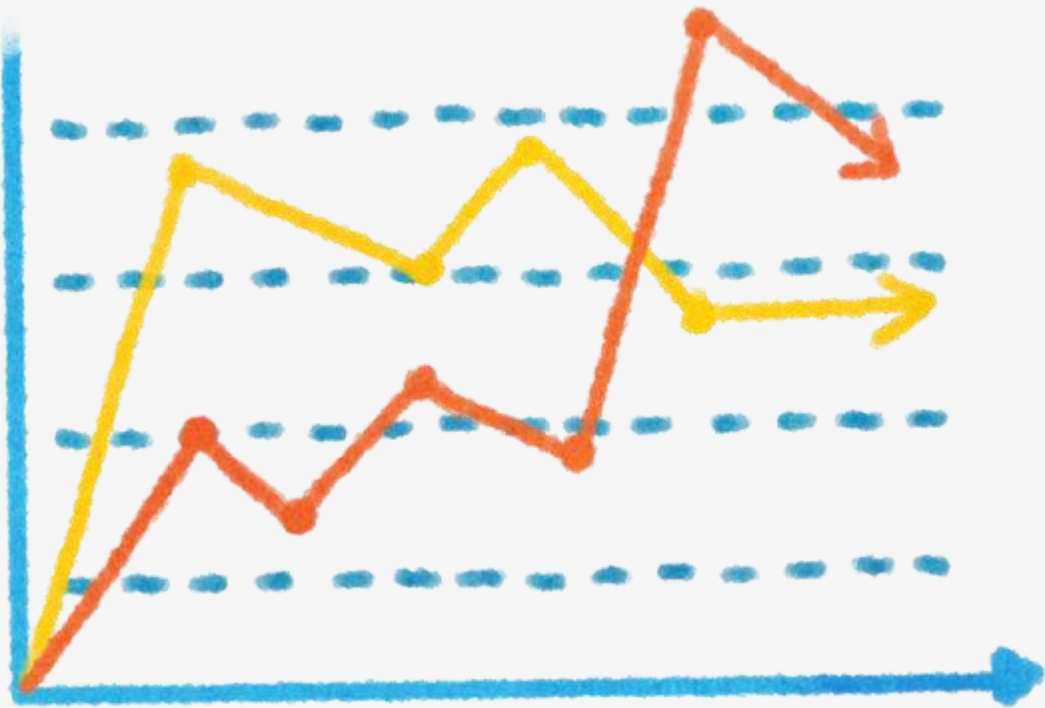
프로젝트 결과 및 평가

- 웹 서비스 기능

1.상품의 실시간 순위 확인



2.상품과 관련한 시각화 자료 제공





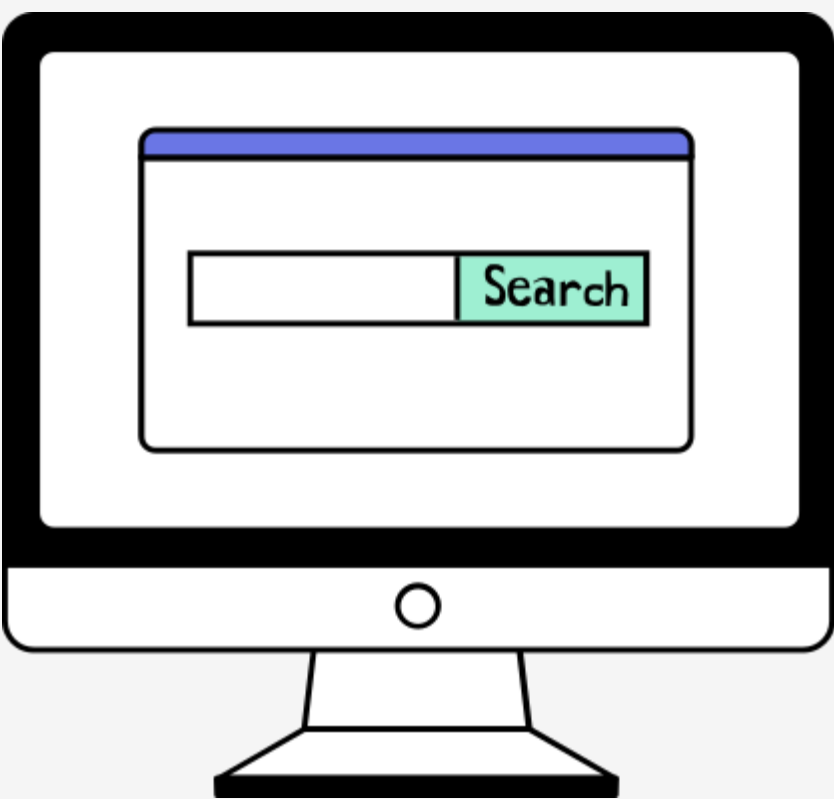
프로젝트 결과 및 평가

- 웹 서비스 기능

3.리셀 플랫폼 기사 확인



4.상품의 가격 예측 서비스





프로젝트 결과 및 평가

- 결론

본 프로젝트에서는 희소성이 높은 상품의 가격이 시간이 지남에 따라 증가하고 있음을 확인하였고, 세부적으로 어떤 브랜드의 상품이 많이 거래가 되고 높은 가격대를 형성하는지 알아봄을 통해 리셀플랫폼 트렌드를 파악할 수 있었다.

고객의 브랜드 및 상품의 가격을 쉽게 예측하여 이용자에게 필요한 정보를 제공하고 본 프로젝트의 웹서비스를 통해 신규 고객을 확보하는 마케팅 전략 수행에 활용할 수 있다.



프로젝트 결과 및 평가

- 프로젝트 평가 및 회고

1. 트렌디한 아이디어로 흥미로운 프로젝트 진행
2. 고객들이 편리하게 상품의 정보를 알 수 있고 깔끔한 UI로 사용하기 쉬운 웹 서비스 개발
3. db를 웹사이트와 연동시켜 대시보드 개선 필요
4. 추후 Heroku 배포 재시도
5. 모델 일반화와 성능 향상을 위해 타 브랜드 데이터 및 최근 2년간의 가격 변화 데이터수집 필요
6. 데이터 수집의 어려움
 - 새로고침 되는 스크롤 웹사이트 크롤링 기술 부족 및 stockx 크롤링 접근 거부
 - 수집한 데이터들의 Feature 유사성 및 일치성 부족 -> 모델 성능 하락
 - 절대적인 데이터 양 부족

The background features a light gray world map composed of small dots. In the top left, there are four green dots in a horizontal line and a green 'X' below them. In the top right, there is a green 'X' and a green circle with a white center. A solid green horizontal band spans the middle of the image, containing the text 'Thank you :)'. On the left side of this band, there is a green rectangle with white diagonal lines. On the right side, there is a green rectangle with white diagonal lines. At the bottom right, there are four green dots in a horizontal line.

Thank you :)