수상작 리뷰 보고서(11/16)

태그

[해외 부동산 월세 예측 AI 해커톤]

https://dacon.io/competitions/official/236044/codeshare/7361?page=1&dtype=recent

1. 데이터

칼럼명

- ID
- propertyType
- bedrooms
- latitude
- longitude
- suburbName
- distanceMetro(km)
- distanceAirport(km)
- distanceHospital(km)
- distanceRailway(km)
- area(square_meters)
- monthlyRent(us_dollar)

2. 코드 흐름

(1) 라이브러리 불러오기

```
import random
import os
import sys

import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

import sklearn
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_absolute_error
from sklearn.preprocessing import OneHotEncoder

from supervised.automl import AutoML

import warnings
warnings.filterwarnings(action='ignore')
```

(2) 시드 고정

```
def seed_everything(seed):
    random.seed(seed)
    os.environ['PYTHONHASHSEED'] = str(seed)
    np.random.seed(seed)

seed_everything(38) # Seed 고정
```

(3) EDA 진행 및 결측치 확인

```
# EDA용 df 데이터셋 생성
total_df = train.drop(columns = ['ID'])
#qualitative: 질적 변수
qual_df = total_df[['propertyType', 'suburbName']]
#quantitative: 양적 변수
quan_df = total_df.drop(columns = ['propertyType', 'suburbNam
```

```
train.isnull().sum()
```

→ 결측치 없음.

(4) 양적 변수

```
# 양적변수 기초통계량 확인
total_df.describe()
#양적 변수 분포 시각화
quan_df.hist(bins=100, figsize=(18,18))
plt.show()
```

(5) 질적 변수

```
#질적 변수 빈도 시각화
fig, axes = plt.subplots(2, 1, figsize=(20,15))

sns.countplot(x = qual_df['propertyType'], ax=axes[0])
sns.countplot(x = qual_df['suburbName'], ax=axes[1])

plt.show()
```

(6) 전처리 진행

```
# ID와 타겟 컬럼 제거
x_train = train.drop(columns=['ID', 'monthlyRent(us_dollar)']
y_train = train['monthlyRent(us_dollar)']
x_test = test.drop(columns=['ID'])
# North Delhi -> Delhi North 로 통일
# West Delhi -> Delhi West 로 통일
x_train.loc[x_train['suburbName']=='North Delhi','suburbName'
x_train.loc[x_train['suburbName']=='West Delhi','suburbName']=
x_test.loc[x_test['suburbName']=='North Delhi','suburbName']=
x_test.loc[x_test['suburbName']=='West Delhi','suburbName']='
```

```
# qualitative column one-hot encoding
qual_col = ['propertyType','suburbName']
ohe = OneHotEncoder(sparse=False)

for i in qual_col:
    x_train = pd.concat([x_train, pd.DataFrame(ohe.fit_transf

    for qual_value in np.unique(x_test[i]):
        if qual_value not in np.unique(ohe.categories_):
            ohe.categories_ = np.append(ohe.categories_, qual
    # One Hot Encoder가 Test 데이터로부터 Fitting되는 것은 Data Le
    x_test = pd.concat([x_test, pd.DataFrame(ohe.transform(x_
```

(7) 로그 정규화

```
x_train.loc[:,:'area(square_meters)'] = np.log1p(x_train.loc[
x_test.loc[:,:'area(square_meters)'] = np.log1p(x_test.loc[:,
```

(8) 모델링(AutoML)

```
# 검증옵션
val_strategy = {
    'validation_type' : 'kfold',
    'k_folds' : 5,
    'shuffle' : True,
    "stratify": True
}
# automl 모델 생성
automl = AutoML(mode = 'Compete', eval_metric='mae', ml_task=
automl.fit(x_train, y_train)
# 생성된 모델로 test 예측하기
pred = automl.predict(x_test)
submission['monthlyRent(us_dollar)'] = pred
# 제출파일 생성
submission.to_csv('./result_20221223_log.csv', index=False)
```

3. **차별점, 배울점**

- 다양한 시각화를 방법을 통해 적절한 데이터 처리를 함. 예를 들어, heatmap, barplot 등을 이용해서 데이터를 확인한 후 결측치를 처리하고 변수를 통일함.
- 로그 정규화를 진행함. 데이터가 정규분포를 따르지 않을 수 있는데 정규화를 통해 모델을 적용하는데 적절한 정규분포를 따르는 형태로 변환함. 이를 통해 좀 더 정확한 결과를 만드는데 기여할 수 있었음.
- automl 머신러닝 모델의 개발 및 최적화 과정을 자동화한 기술이나 도구를 의미함. 일반적으로 머신러닝 모델을 만들려면 데이터 전처리, 특성 엔지니어링, 모델 선택, 하이퍼파라미터 튜닝, 그리고 평가 과정을 거쳐야 하는데, 이 과정은 많은 시간이 들고, 전문 지식이 요구됨. 하지만, AutoML은 이 과정을 자동화하여, 비전문가도 쉽게 고품질의 모델을 만들 수 있도록 도움. 이를 통해서 정확도가 높은 결과를 산출할 수 있었음.