

3 조 진행상황

차원축소

스케일러의 종류에 따라 결과가 다르게 나오는지 확인

→ 압축 성능 좋지 않아 적용하지 않기로 결정

- Robust Scaler(정확도 : 0.1019)

로버스트스케일링+차원축소

```
In [8]: #PCA차원축소
from sklearn.decomposition import PCA
from sklearn.preprocessing import RobustScaler
from sklearn.model_selection import cross_val_score
#원본스케일링
scaler = RobustScaler()
df_scaled=scaler.fit_transform(X_train)
pca=PCA(n_components=6)
df_pca = pca.fit_transform(df_scaled)

In [10]: #정확도 판별 - 랜덤포레스트
#기준 세트 정확도 80%
rcf=RandomForestClassifier(n_estimators=300, random_state=156)
scores_pca = cross_val_score(rcf,df_pca, y_train, scoring='accuracy', cv=3)
print('CV3인 경우 PCA변환된 개별 fold 세트별 정확도 : ', scores_pca)
print('PCA 변환 데이터 세트 평균 정확도 : {0:.4f}'.format(np.mean(scores_pca)))

CV3인 경우 PCA변환된 개별 fold 세트별 정확도 : [0.10266407 0.09953637 0.10346842]
PCA 변환 데이터 세트 평균 정확도 : 0.1019
```

- MinMax Scaler(정확도 : 0.3223)

```
In [9]: #정확도 판별 - 랜덤포레스트
#기준 세트 정확도 80%
error_score='raise'
rcf=RandomForestClassifier(n_estimators=300, random_state=156)
scores_pca = cross_val_score(rcf,df_pca, y_train, scoring='accuracy', cv=3)
print('CV3인 경우 PCA변환된 개별 fold 세트별 정확도 : ', scores_pca)
print('PCA 변환 데이터 세트 평균 정확도 : {0:.4f}'.format(np.mean(scores_pca)))

CV3인 경우 PCA변환된 개별 fold 세트별 정확도 : [0.31680023 0.32048682 0.32975539]
PCA 변환 데이터 세트 평균 정확도 : 0.3223
```

- Standard Scaler(정확도 : 0.6526)

#차원축소 분류 예측 성능 평가

#원본 데이터셋

#랜덤 포레스트 이용해 타깃 값이 디폴트 값을 3개 교차 검증 세트로 분류 예측

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import cross_val_score
```

```
rclf=RandomForestClassifier(n_estimators=300, random_state=156)
scores = cross_val_score(rclf, X_train, y_train, scoring='accuracy', cv=3)
print('CV3인 경우 개별 fold 세트별 정확도 : ', scores)
print('평균 정확도 : {0:.4f}'.format(np.mean(scores)))
```

CV3인 경우 개별 fold 세트별 정확도 : [0.7795827 0.80259345 0.79941585]
평균 정확도 : 0.7939

#PCA차원축소

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

#원본스케일링

```
scaler = StandardScaler()
df_scaled=scaler.fit_transform(X_train)
```

#컴포넌트 임의 6개 선정

```
pca=PCA(n_components=6)
df_pca = pca.fit_transform(df_scaled)
scores_pca = cross_val_score(rclf,df_pca, y_train, scoring='accuracy', cv=3)
print('CV3인 경우 PCA변환된 개별 fold 세트별 정확도 : ', scores_pca)
print('PCA 변환 데이터 세트 평균 정확도 : {0:.4f}'.format(np.mean(scores_pca)))
```

CV3인 경우 PCA변환된 개별 fold 세트별 정확도 : [0.64587394 0.64872501 0.66316174]
PCA 변환 데이터 세트 평균 정확도 : 0.6526

단계

데이터로드 → lgbm Bayesian Optimization → 하이퍼파라미터튜닝 → 모델생성

결과

SCORE 0.72723 (#42)

수정

```
pbounds = {'learning_rate' : (0.01, 0.3),  
           'n_estimators' : (50, 1000),  
           'num_leaves' : (24, 45),  
           'feature_fraction' : (0.1, 0.9),  
           'bagging_fraction' : (0.8, 1),  
           'max_depth' : (5, 8.99),  
           'lambda_l1' : (0, 5),  
           'lambda_l2' : (0, 3),  
           'min_split_gain' : (0.001, 0.1),  
           'min_child_weight' : (5, 50)  
          }
```

검토

양상블 - 탐색 파라미터를 3 개의 모델에 할당하여 모델링 후 평균값 사용

결과 2

SCORE 1.2255148803
