머신러닝 – 랜덤포레스트 분류 분석

1. 랜덤포레스트 적용 전 전처리

- 독립변수 비용 열 2개(expense, expense_f) 범주 축소
- 종속변수 pd.cut() 사용하여 카테고리화
- 시니어 및 수도권 필터링 후 열 삭제

2. 랜덤포레스트 모델 생성

```
# 랜덤포레스트 모델 생성
from sklearn.ensemble import RandomForestClassifier

rf_clf = RandomForestClassifier(random_state=0, n_estimators=300)

rf_clf.fit(X_train, y_train)
y_pred = rf_clf.predict(X_test)

print('랜덤 포레스트 예측 정확도 : {0:.4f}'.format(accuracy_score(y_test, y_pred)))
```

```
RandomForestClassifier(n_estimators=300, random_state=0)
```

랜덤 포레스트 예측 정확도 : 0.5404

3. 하이퍼 파라미터 튜닝 후 재 학습 및 예측

튜닝된 하이퍼 파라미터로 다시 학습/예측/평가 수행

```
rf_clf = RandomForestClassifier(n_estimators=200,
                                max_depth=7,
                                min_samples_leaf=2,
                                min_samples_split=4)

rf_clf.fit(X_train, y_train)
y_pred = rf_clf.predict(X_test)

print('최적 정확도 : {0:.4f}'.format(accuracy_score(y_test, y_pred)))
```

```
RandomForestClassifier(max_depth=7, min_samples_leaf=2, min_samples_split=4,
                       n_estimators=200)
```

최적 정확도 : 0.5466

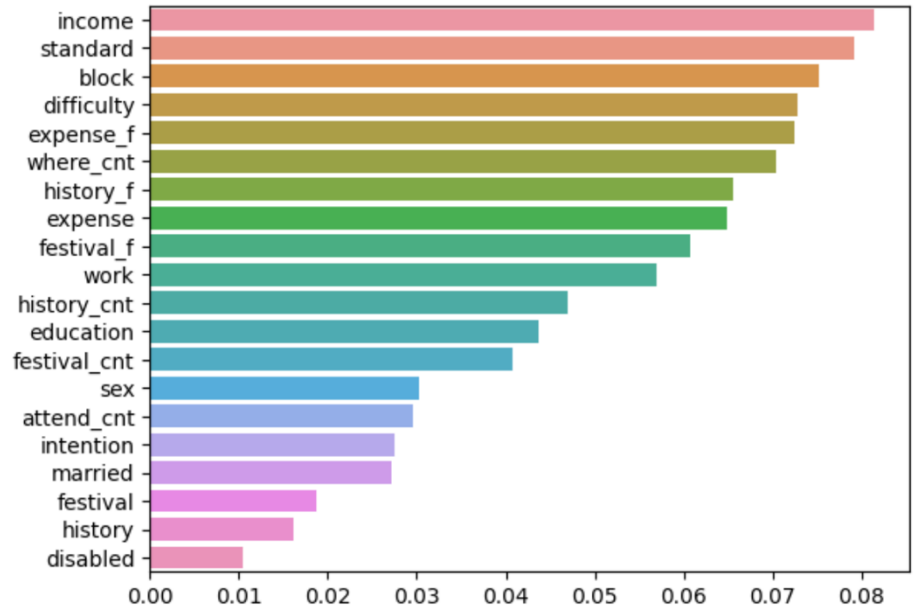
4. 랜덤포레스트 모델의 Feature 중요도 시각화

개별 feature들의 중요도 시각화

```
# 피처 중요도가 높은 20개의 피처만
f_imp = rf_clf.feature_importances_
feature_importances = pd.Series(f_imp, index=X_train.columns).sort_values(ascending=False)
feature_importances = feature_importances[:20]

sns.barplot(x=feature_importances, y=feature_importances.index)
```

```
<Axes: >
```



가장 변별력이 높은 feature는 소득(income)이다.