Forest_top5 함수 설명

```
def forest_top5(data, fnames, k):

  from sklearn.ensemble import RandomForestRegressor
  from sklearn.metrics import mean_squared_error
  from sklearn.metrics import r2_score
  import numpy as np

  y_train = data[(data['발생년도']<2018)]['사고건수']
  y_test = data[(data['발생년도']==2018)]['사고건수']
  y_test= y_test.reset_index(drop=True)

  from itertools import combinations as cm

  scores=[]


  fsets=list(cm(fnames,k)) #변수명에 대해 nCk의 조합

  for cmb in fsets: #각 조합에 대해
    fset=[]
    for x in cmb:
        fset.append(x)


    X_train = data[(data['발생년도']<2018)][fset] #train, test셋 만들고
    X_test = data[(data['발생년도']==2018)][fset]

    forest = RandomForestRegressor(n_estimators=100, #모델 적합
                                   criterion='squared_error',
                                   random_state=1,
                                   n_jobs=-1)
```

y_train 에는 발생년도가 2018년 이전의 사고건수 데이터를,
y_test에는 발생년도가 2018년인 사고건수 데이터를 지정합니다.

combination 함수를 사용하기 위한 모듈을 불러옵니다.

fsets라는 리스트에 fnames 리스트 안에 있는 변수에 대해 k개를 뽑는 조합(nCk)을 원소로 갖는 리스트를 생성해줍니다.

데이터셋에서 각각 fset에 해당하는 특성만 추려내서 X_train, X_test 데이터셋을 만들어주도록 합니다.

랜덤 포레스트 분류기를 만들어줍니다.

Forest_top5 함수 설명

```
  forest.fit(X_train, y_train)          #적합

  y_train_pred = forest.predict(X_train) #예측
  y_test_pred = forest.predict(X_test)

  print('₩n 랜덤포레스트 변수: ', fset)  #해당 모델에 사용된 변수 조합

  print('훈련 MSE: %.3f, 테스트 MSE: %.3f' % (
    mean_squared_error(y_train, y_train_pred),
    mean_squared_error(y_test, y_test_pred)))
  print('훈련 R^2: %.3f, 테스트 R^2: %.3f' % (
        r2_score(y_train, y_train_pred),
        r2_score(y_test, y_test_pred)))

  scores.append( (round(r2_score(y_test, y_test_pred),3),fset) ) #for문 안에서 튜플 추가

print(sorted(scores, key=lambda x: x[1], reverse=True)[:5]) #성적순 정렬해 상위 점수 5개 조합 반환
```

각각 X_train, X_test 데이터로 forest 분류기로 예측한 값을 y_train_pred와 y_test_pred 변수에 넣어줍니다.

y_train, y_test 데이터 각각의 mean_squared_error를 출력해줍니다.

y_train, y_test 데이터 각각의 r2_score를 출력해줍니다.

scores 리스트에 fset의 r2_score를 소수 3째짜리까지 반올림한 값을 요소로 추가해줍니다.

마지막으로, score 순서대로 정렬하여 상위 점수 5개의 조합을 반환하도록 합니다.

```
 랜덤포레스트 변수:  ['노면상태', '도로형태']
훈련 MSE: 0.025, 테스트 MSE: 0.064
훈련 R^2: 0.817, 테스트 R^2: 0.541

 랜덤포레스트 변수:  ['노면상태', '도로형태_대분류']
훈련 MSE: 0.069, 테스트 MSE: 0.082
훈련 R^2: 0.487, 테스트 R^2: 0.416

 랜덤포레스트 변수:  ['기상상태', '도로형태']
훈련 MSE: 0.004, 테스트 MSE: 0.009
훈련 R^2: 0.970, 테스트 R^2: 0.939

 랜덤포레스트 변수:  ['기상상태', '도로형태_대분류']
훈련 MSE: 0.009, 테스트 MSE: 0.011
훈련 R^2: 0.931, 테스트 R^2: 0.924

 랜덤포레스트 변수:  ['도로형태', '도로형태_대분류']
훈련 MSE: 0.013, 테스트 MSE: 0.033
훈련 R^2: 0.901, 테스트 R^2: 0.765
[(0.939, ['기상상태', '도로형태']), (0.924, ['기상상태', '도로형태_대분류']), (0.946, ['노면상태', '기상상태']), (0.541, ['노면상태', '도로형태']), (0.416, ['노면상태', '도로형태_대분류'])]
```

다음과 같이 출력되는 것을 확인할 수 있습니다.