Code

Out[44]: array([0.65270969, 0.64576405, 0.64861308, 0.64370606, 0.6461396])

In [ ]:
```python
log_pipe.fit(X_train,y_train)
jobsfinished()
```

In [ ]:

In [63]:
```python
#DT Grid Search  Mk2


#,10,20   #, 16, 30
params = {'tree__max_depth': [10], 'tree__min_samples_split': [8], 'tree__min_samples_leaf':[3,10]

         }
grid_search_cvMk2 = GridSearchCV(dt_pipe, param_grid=params, verbose=2, cv=5, n_jobs=-3, scoring="f1_weighted")
grid_search_cvMk2.fit(X_train, y_train)

print(grid_search_cvMk2.best_estimator_)
print(grid_search_cvMk2.best_score_)
```

▮▮ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ -0:02

Fitting 5 folds for each of 2 candidates, totalling 10 fits

```
[Parallel(n_jobs=-3)]: Using backend LokyBackend with 10 concurrent workers.
[Parallel(n_jobs=-3)]: Done   3 out of  10 | elapsed:  2.7min remaining:  6.2min
[Parallel(n_jobs=-3)]: Done  10 out of  10 | elapsed:  3.4min finished
```

▮▮ ▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬▬ -0:01

```
Pipeline(steps=[('preprocess',
                 ColumnTransformer(remainder='passthrough',
                                   transformers=[('subpipe_num',
                                                  Pipeline(steps=[('num_impute',
                                                                   SimpleImputer()),
                                                                  ('ss',
                                                                   StandardScaler())]),
                                                  ['gps_height', 'population']),
                                                 ('subpipe_cat',
                                                  Pipeline(steps=[('cat_impute',
                                                                   SimpleImputer(strategy='most_frequent')),
                                                                  ('ohe',
                                                                   OneHotEncoder(handle_unknown='ignore',
                                                                                 sparse=False))]),
                                                  ['funder', 'installer',
                                                   'basin', 'subvillage'])])),
                ('tree',
                 DecisionTreeClassifier(class_weight='balanced', max_depth=10,
                                        min_samples_leaf=3, min_samples_split=8,
```