implemented clustering algorithm

- **k-means**

```
'''
k: number of clusters
max_iter: max iteration number
metric: distance metric between two samples
'''
_kmeans = kmeans(k = 3, max_iter = 1000, metric = 'euclidean')
out = _kmeans.fit(X)
```

- **kmedoids**

```
'''
k: number of clusters
metric: distance metric between two samples
n_trials: kmedoids run times
max_iter: maximum number of iterations in a trial
tol: tolerance
'''
_kmedoids = kmedoids(k = 3, metric = 'euclidean' ,n_trial = 10, max_iter = 1000,
tol = 0.001)
out = _kmedoids.fit(X)
```

- **kmeans++**

```
'''
k: number of clusters
maxiter: max iteration number
metric: distance metric between two samples
'''
_kmeanspp = Kmeanspp(k = 3, max_iter = 1000, metric = 'euclidean')
out = _kmeanspp.fit(X)
```

- **Agglomerative Hierarchical Clustering**

```
'''
implement from sklearn
also implement the cluster merging, mentioned in paper(NSDI22, CloudCluster)

aff: metric used to compute the linkage
link: the linkage criterion determines which distance to use between sets of observation
dist_threshold: distance_threshold, the linkage distance threshold above which, clusters will not be merged
'''
ah_cluster = agg_clustering(aff = 'euclidean', link = 'ward', dist_threshold = 3.0, n_cluster = None)
out = ah_cluster.fit(X)
merged_out = ah_cluster.cluster_merge(X, out, threshold=0.001)
```