t-SNE i UMAP

t-SNE i UMAP su algoritmi za smanjivanje dimenzija podataka koji se oslanjaju na algebarsku topologiju prilikom analize podataka, određujući utapanje koje čuva topologiju.

Ako u skupu podataka imamo na primer 50 atributa, cilj je da, koristeći t-SNE ili UMAP, preslikamo prostor $\mathbb{R}^{50}$ u neki prostor manje dimenzije. Ove metode se često koriste za vizualizaciju pa preslikavamo u prostor $\mathbb{R}^2$ ili $\mathbb{R}^3$ jer ih je moguće nacrtati.

Ideja iza ove dve metode da se tačke preslikaju u manjedimenzioni prostor tako da se sačuvaju odnosi rastojanja. Odnosno, cilj je da tačke koje su bile blizu u početnom prostoru, budu blizu i u novom prostoru, a da one koje su bile originalno daleko tako i ostanu. Dakle, nije cilj da se očuva rastojanje, već susedstvo.

Ukratko ćemo objasniti t-SNE. Prvo je potrebno definisati kako računamo sličnost dve instance. To će biti neka funkcija od norme njihove razlike. Logično je da je norma razlike dve slične instance mala, a da je norma razlike dve različite insatnce velika. Izbor funkcije i norme može biti različit. Označimo sada sa p_{ij} sličnost i-te i j-te instance u originalnom prostoru, a sa q_{ij} njihovu sličnost u novom prostoru.

Funkcija gubitaka je Kulbak-Lajblerovo rastojanje, koje je definisano sa:

$$L = \sum_{i,j} p_{ij} log \frac{p_{ij}}{q_{ij}}$$

Iz formule se vidi da se plaća velika cena ako se bliske tačke projektuju daleko. Funkcija gubitaka se minimizuje metodom gradijentnog spusta.
Za detaljnije objašnjenje pogledati sledeći snimak: https://www.youtube.com/watch?v=NEaUSP4YerM

Za objašnjenje UMAP algoritma pogledati sledeće snimke:
https://www.youtube.com/watch?v=eN0wFzBA4Sc
https://www.youtube.com/watch?v=jth4kEvJ3P8
Za sve detalje pogledati dokumentaciju na sledećem $linku$

U radu ćemo koristiti Viskonsis skup podataka za klasifikaciju tumora na benigne i maligne. Dakle, imamo dve različite klase.

In [3]:
```
# !pip install umap-learn
```

In [4]:
```
import numpy as np
import pandas as pd
```

```
import matplotlib.pyplot as plt
import umap.umap_ as umap
from sklearn import datasets
from sklearn.manifold import TSNE
```

In [5]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [11]:
```
podaci = datasets.load_breast_cancer()
```

In [15]:
```
y = podaci.target
X = podaci.data
```

In [16]:
```
X.shape # originalni prostor je dimenzije 30
```

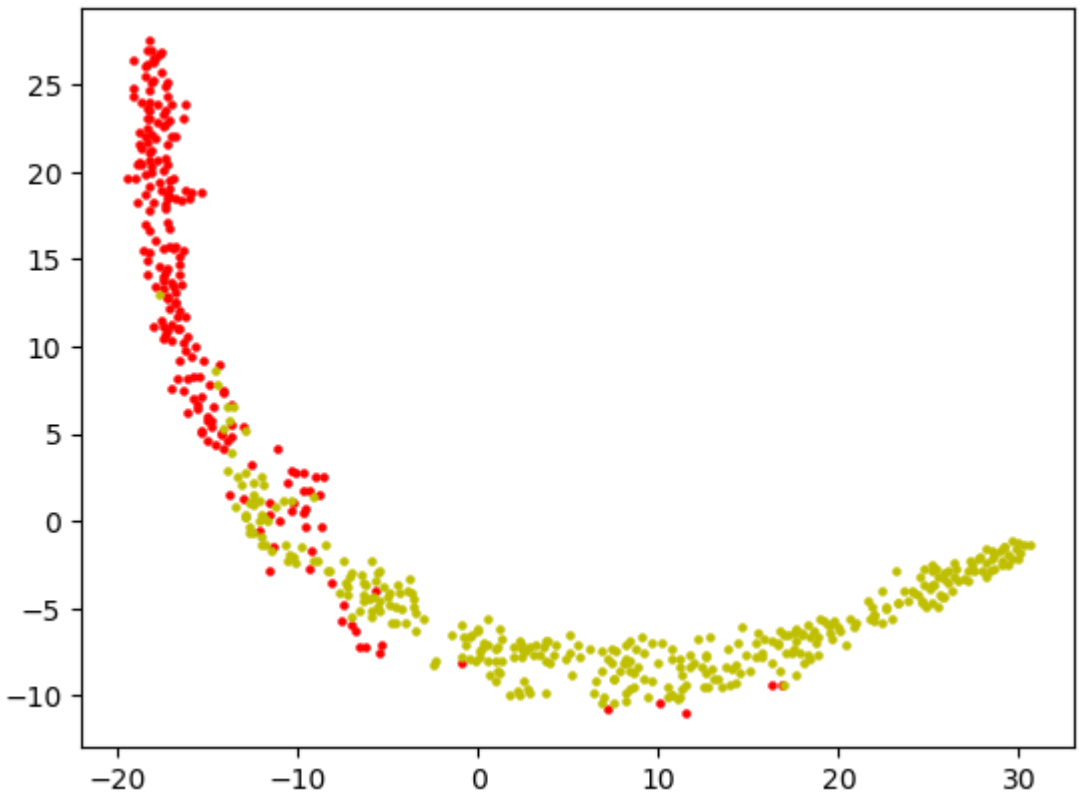
Out[16]:
```
(569, 30)
```

In [17]:
```
tsne = TSNE(n_components=2, perplexity=50) # novi prostor je dimenzije 2
# perplexity je parametar sa kojim je moguce eksperimentisati,
# razlicite vrednosti ce davati razlicite rezultate
X_tsne = tsne.fit_transform(X)
```

In [18]:
```
X_tsne.shape
```

Out[18]:
```
(569, 2)
```

In [19]:
```
plt.scatter(X_tsne[:,0][y==0], X_tsne[:,1][y==0], s=5, c='r')
plt.scatter(X_tsne[:,0][y==1], X_tsne[:,1][y==1], s=5, c='y')
plt.show()
```



Možemo primetiti da su klase sasvim dobro razdvojene.

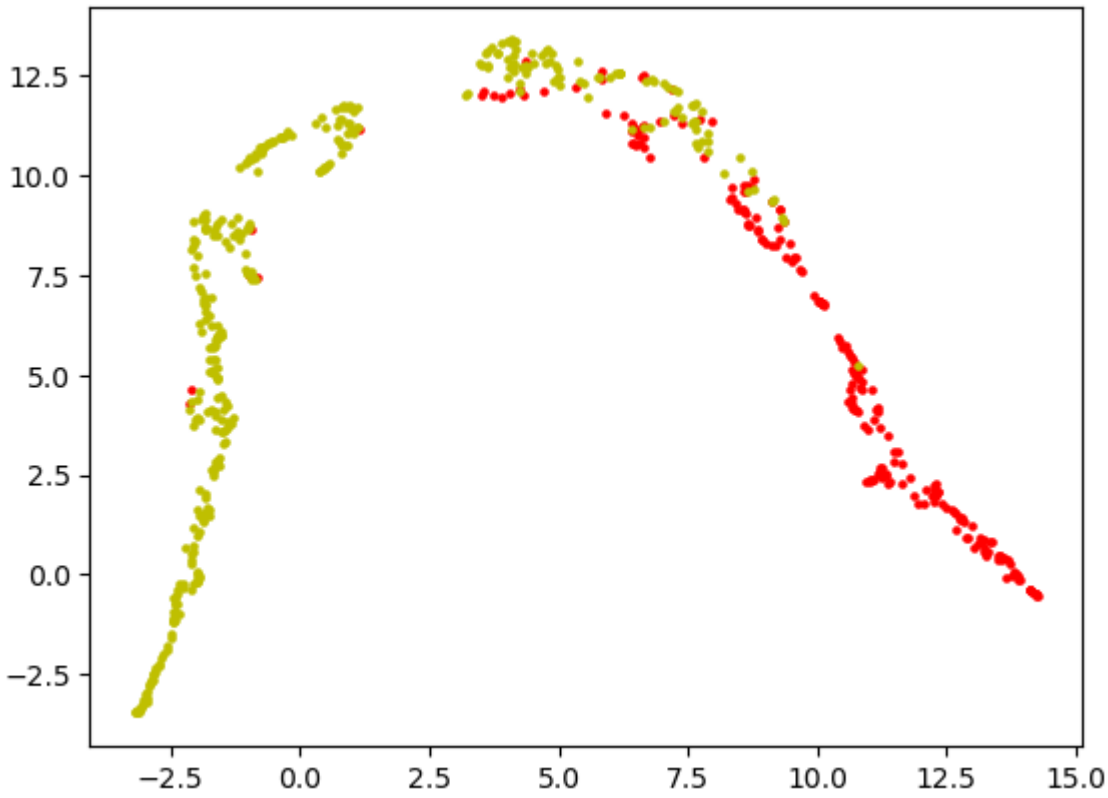
In [20]:
```
umap = umap.UMAP(n_components=2)
X_umap = umap.fit_transform(X)
```

```
OMP: Info #276: omp_set_nested routine deprecated, please use omp_set_max_act
ive_levels instead.
```

In [21]:
```
X_umap.shape
```

Out[21]:
```
(569, 2)
```

In [22]:
```
plt.scatter(X_umap[:,0][y==0], X_umap[:,1][y==0], s=5, c='r')
plt.scatter(X_umap[:,0][y==1], X_umap[:,1][y==1], s=5, c='y')
plt.show()
```



UMAP je dao slične rezultate kao t-SNE.