T-Distributed Stochastic Neighbor Embedding

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.datasets import load_breast_cancer as bc, load_iris as l_iris
from sklearn.preprocessing import StandardScaler as SS
from sklearn.manifold import TSNE

%matplotlib inline
```

In [2]:
```
cancer = bc()                                         ## Instantiating Breast Cancer Datase
iris = l_iris()                                       ## Instantiating IRIS Datset object
ss = SS()                                             ## Instantiating Standard Scaler
tsne = TSNE(n_components=2, random_state=41)          ## Instantiating TSNE
```

In [3]:
```
cancer_df = pd.concat([pd.DataFrame(cancer.data,columns=cancer.feature_names),pd.Dat
cancer_norm_df = pd.DataFrame(ss.fit_transform(cancer_df.iloc[:,0:-1]),columns=cance
cancer_norm_df.head(10)
```

Out[3]:

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	n symm
0	1.097064	-2.073335	1.269934	0.984375	1.568466	3.283515	2.652874	2.532475	2.21
1	1.829821	-0.353632	1.685955	1.908708	-0.826962	-0.487072	-0.023846	0.548144	0.00
2	1.579888	0.456187	1.566503	1.558884	0.942210	1.052926	1.363478	2.037231	0.93
3	-0.768909	0.253732	-0.592687	-0.764464	3.283553	3.402909	1.915897	1.451707	2.86
4	1.750297	-1.151816	1.776573	1.826229	0.280372	0.539340	1.371011	1.428493	-0.00
5	-0.476375	-0.835335	-0.387148	-0.505650	2.237421	1.244335	0.866302	0.824656	1.00
6	1.170908	0.160649	1.138125	1.095295	-0.123136	0.088295	0.300072	0.646935	-0.06
7	-0.118517	0.358450	-0.072867	-0.218965	1.604049	1.140102	0.061026	0.281950	1.40
8	-0.320167	0.588830	-0.184080	-0.384207	2.201839	1.684010	1.219096	1.150692	1.96
9	-0.473535	1.105439	-0.329482	-0.509063	1.582699	2.563358	1.738872	0.941760	0.79

10 rows × 30 columns

In [4]:
```
cancer_df['target'].value_counts()
```

Out[4]:
```
1    357
0    212
Name: target, dtype: int64
```

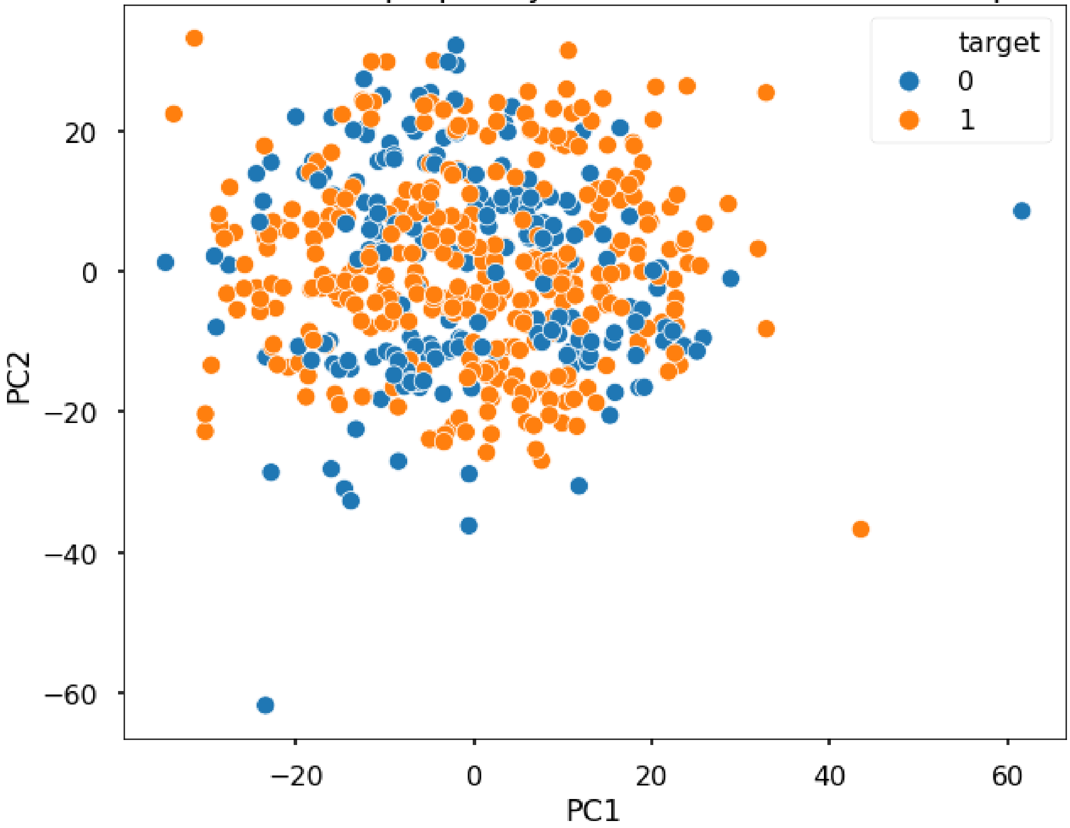
t-SNE_on_boundary_values

Boundary:CASE-I

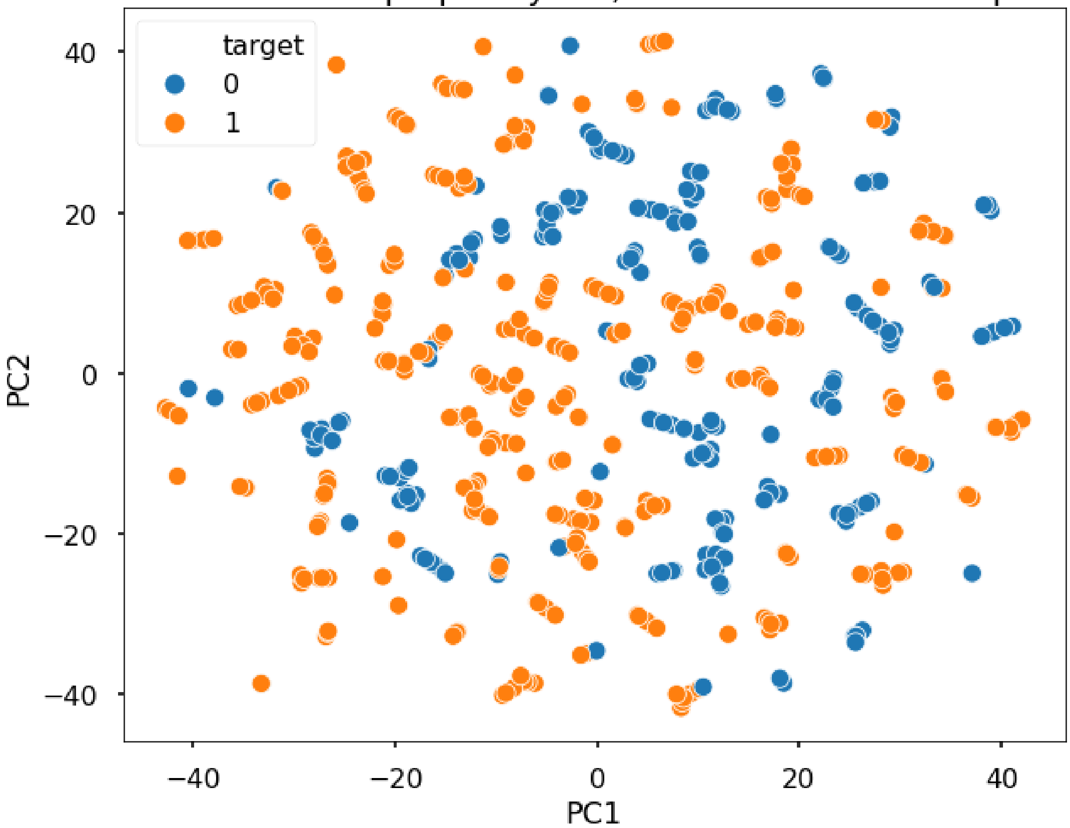
- *### Lowest boundary value of Perplexity as 1 on different Iterations*

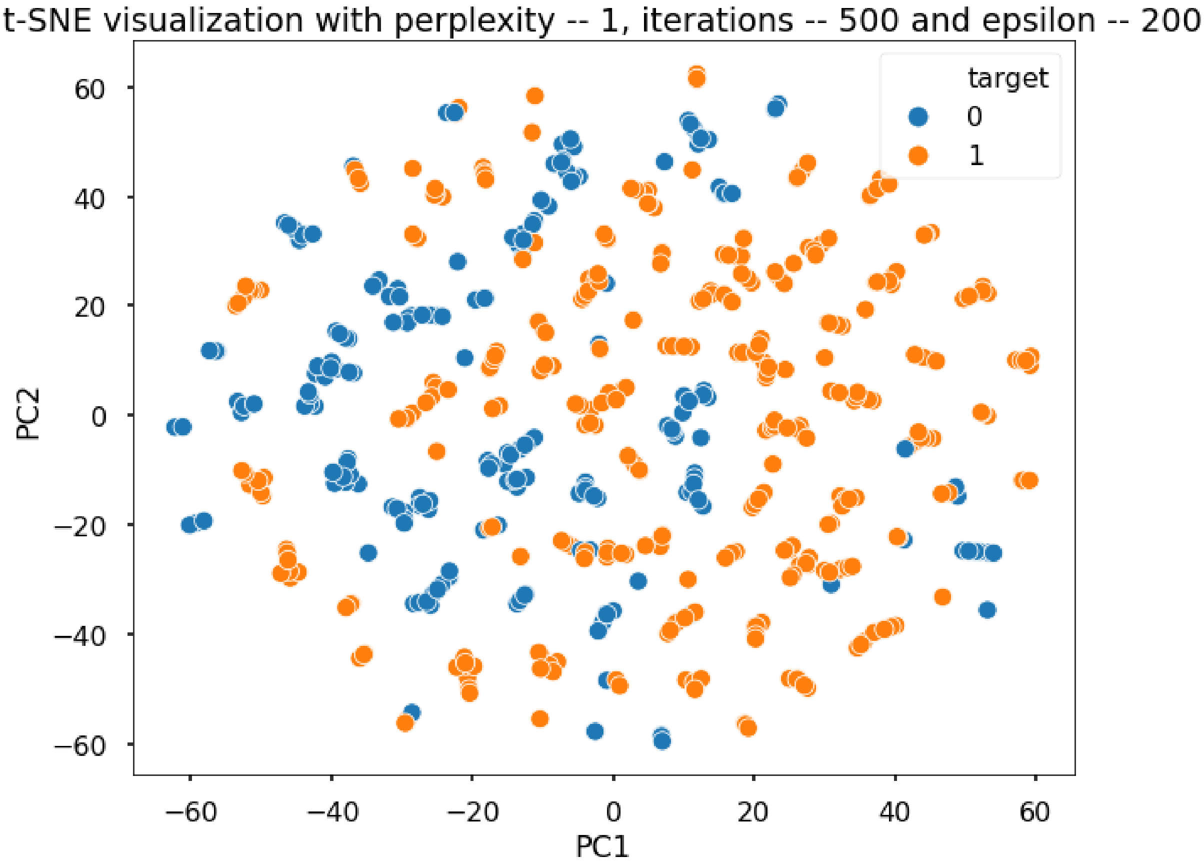
In [83]:
```
iterations = [250,350,500,750,1000,2500,3500,4000,5000]

for idx in range(len(iterations)):
    tsne1 = TSNE(n_components=2,perplexity=1,learning_rate=200,n_iter=iterations[idx
    cancer_tsne_pcmps = pd.DataFrame(tsne1.fit_transform(cancer_norm_df),columns=['P
    cancer_tsne_pcmps = pd.concat([cancer_tsne_pcmps,cancer_df['target']],axis=1)
    with plt.style.context('seaborn-poster'):
        plt.figure(figsize=(7,7))
        sns.scatterplot(data=cancer_tsne_pcmps,x='PC1',y='PC2',hue='target')
        plt.title("t-SNE visualization with perplexity -- {0}, iterations -- {1} and


        plt.show()
```

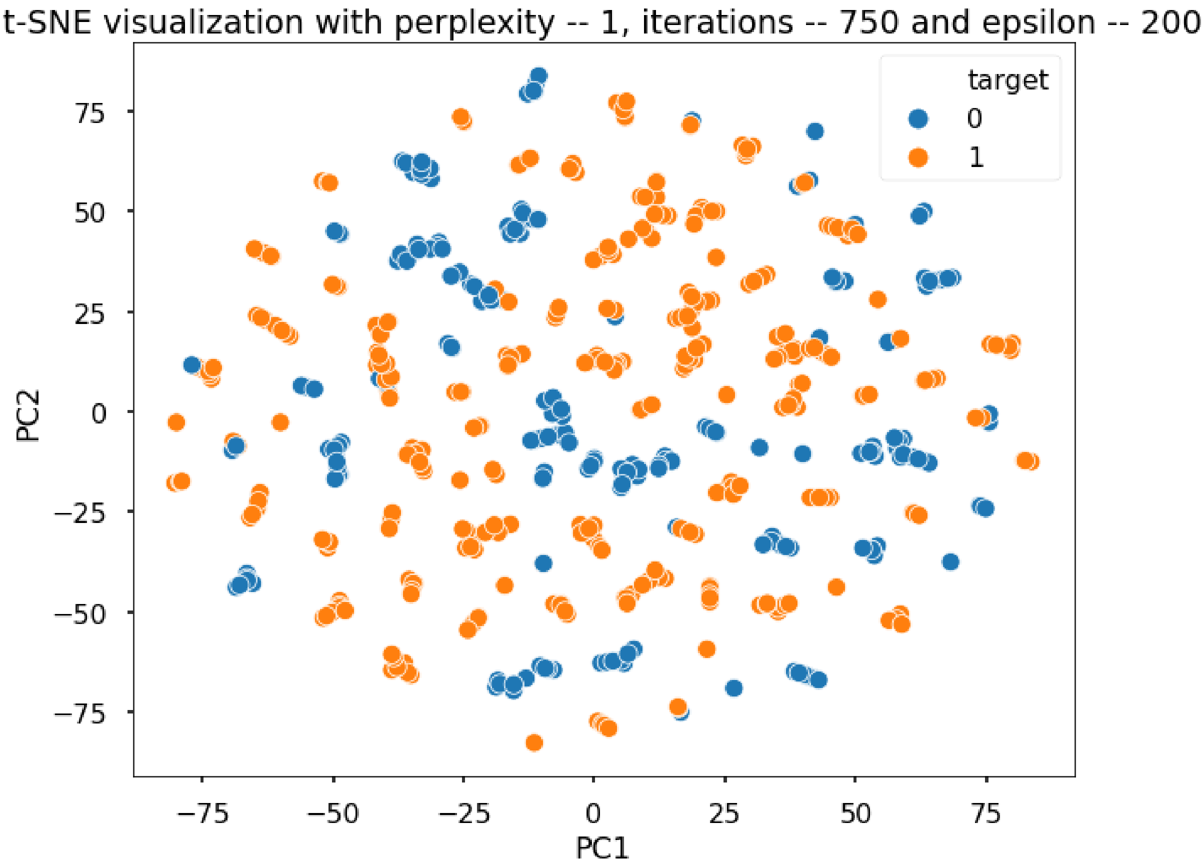
t-SNE visualization with perplexity -- 1, iterations -- 250 and epsilon -- 200

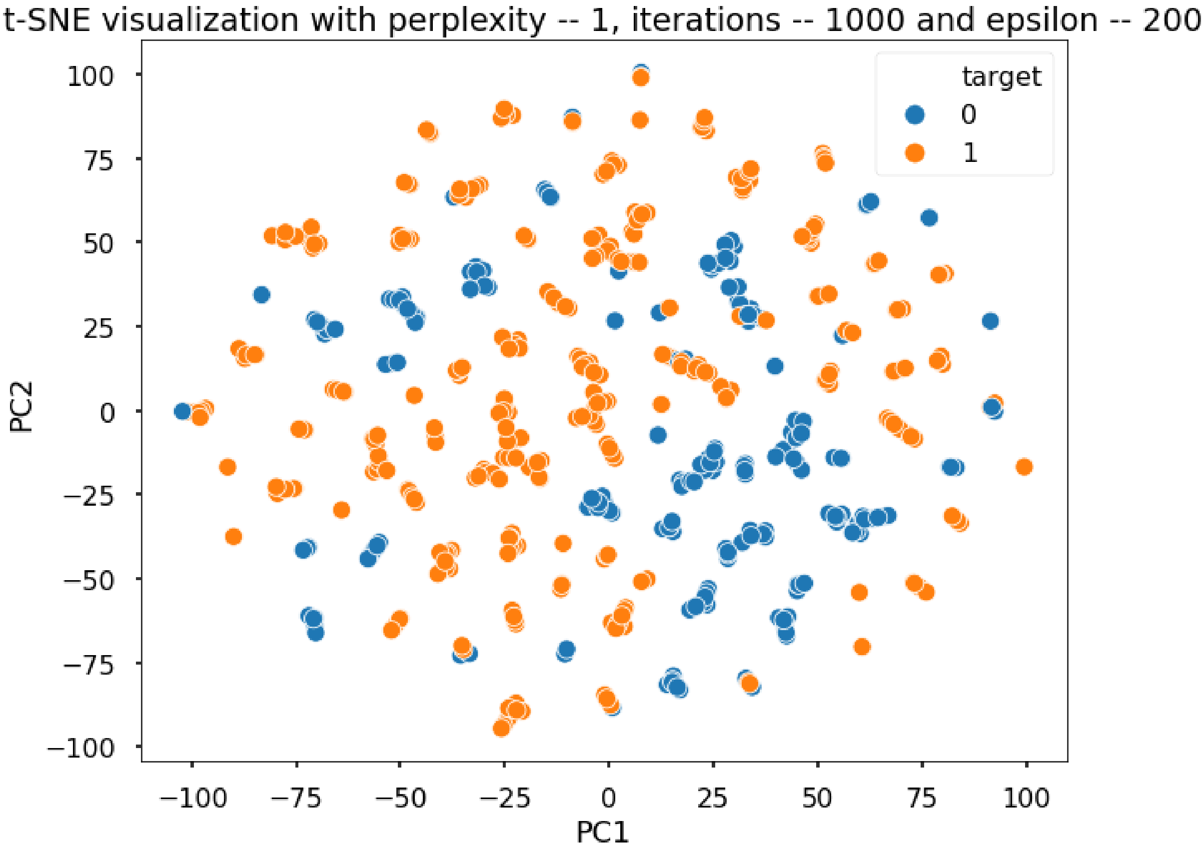


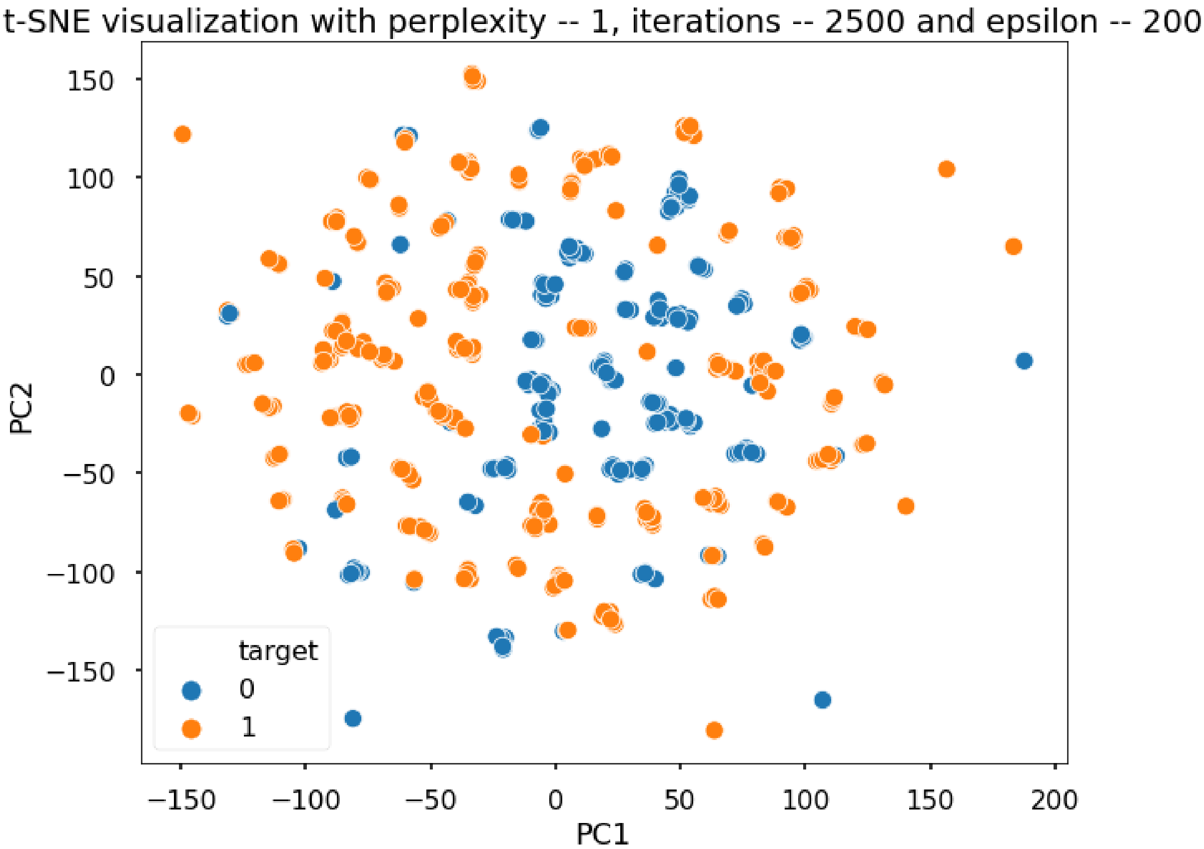
t-SNE visualization with perplexity -- 1, iterations -- 350 and epsilon -- 200



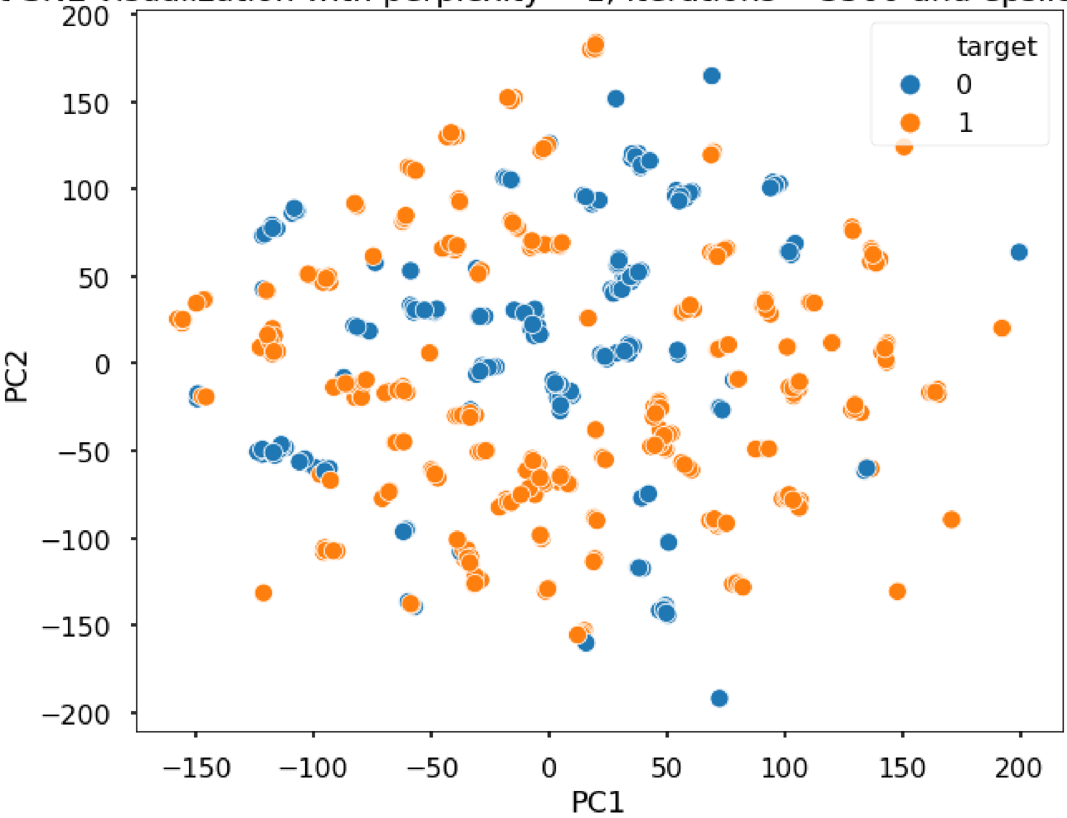

t-SNE visualization with perplexity -- 1, iterations -- 500 and epsilon -- 200
target
0
1
PC2
PC1
60
40
20
0
−20
−40
−60
−60
−40
−20
0
20
40
60


t-SNE visualization with perplexity -- 1, iterations -- 750 and epsilon -- 200
target
0
1
PC2
PC1
75
50
25
0
−25
−50
−75
−75
−50
−25
0
25
50
75

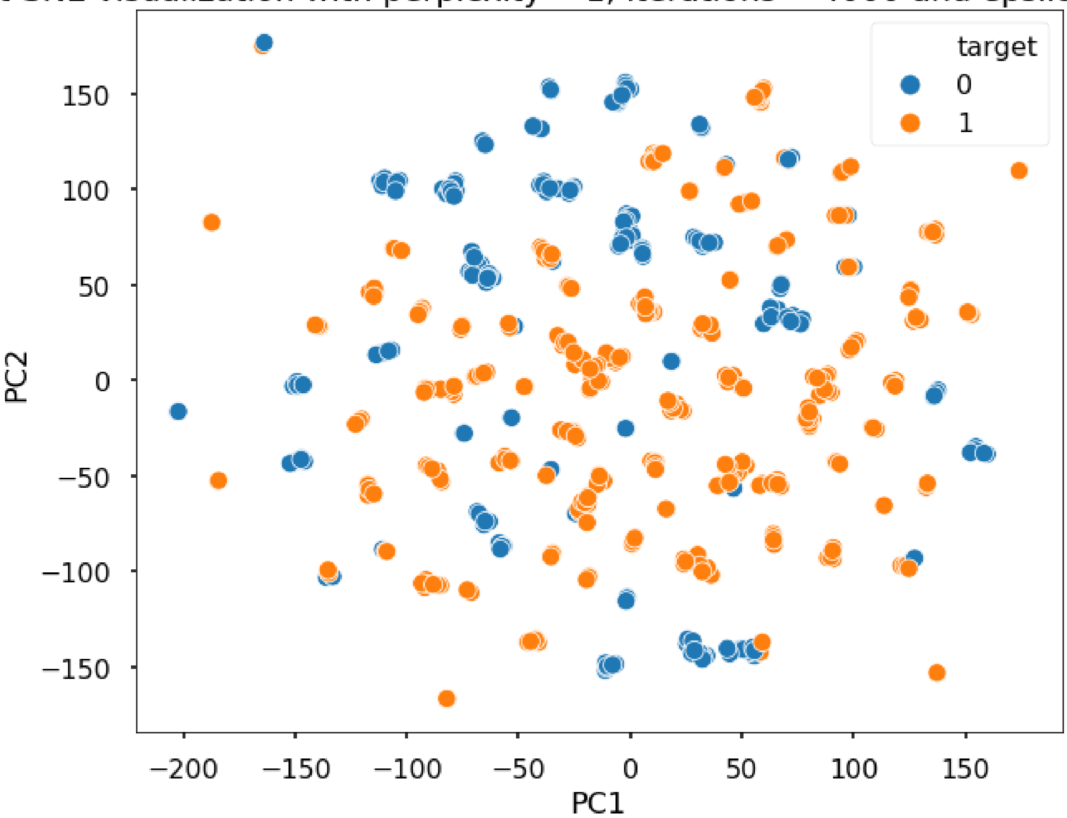

t-SNE visualization with perplexity -- 1, iterations -- 1000 and epsilon -- 200
target
0
1
PC2
PC1
100
75
50
25
0
−25
−50
−75
−100
−100
−75
−50
−25
0
25
50
75
100

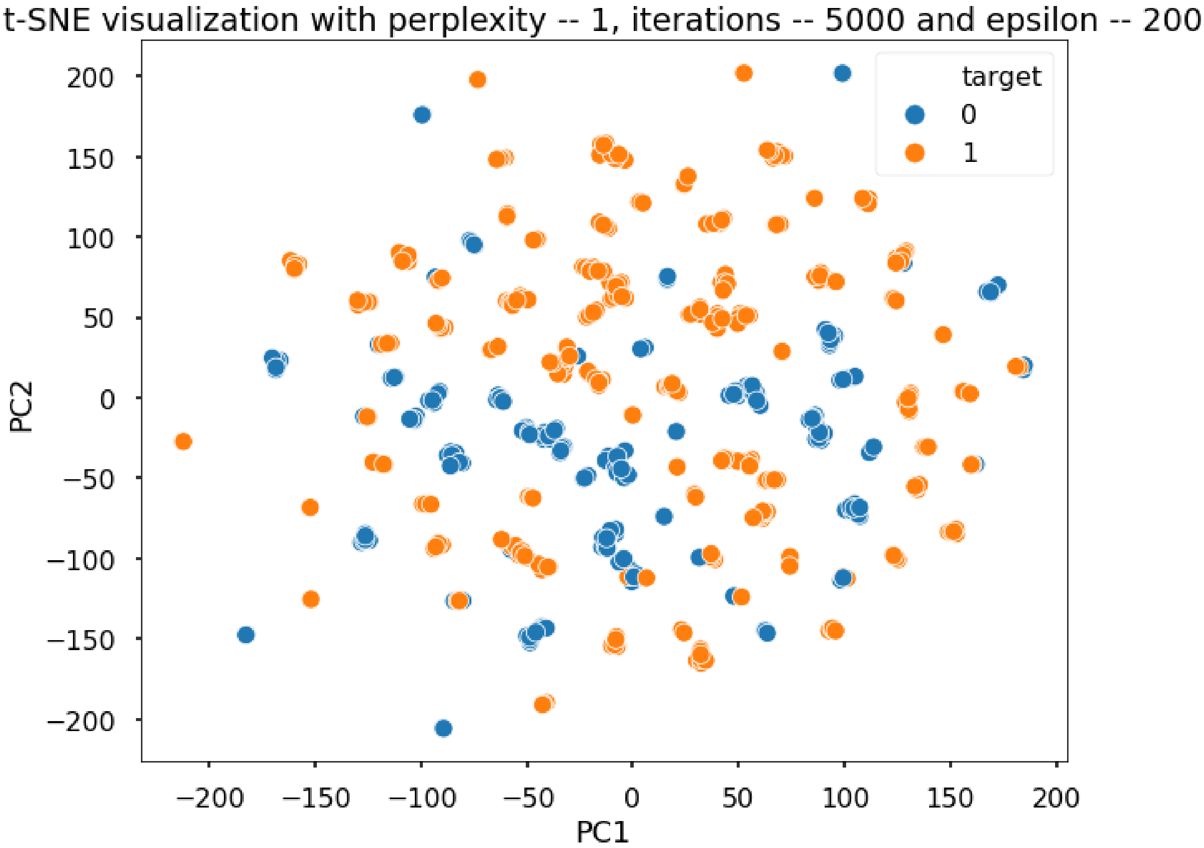

t-SNE visualization with perplexity -- 1, iterations -- 2500 and epsilon -- 200
target
0
1
PC2
PC1
150
100
50
0
−50
−100
−150
−150
−100
−50
0
50
100
150
200

t-SNE visualization with perplexity -- 1, iterations -- 3500 and epsilon -- 200



t-SNE visualization with perplexity -- 1, iterations -- 4000 and epsilon -- 200



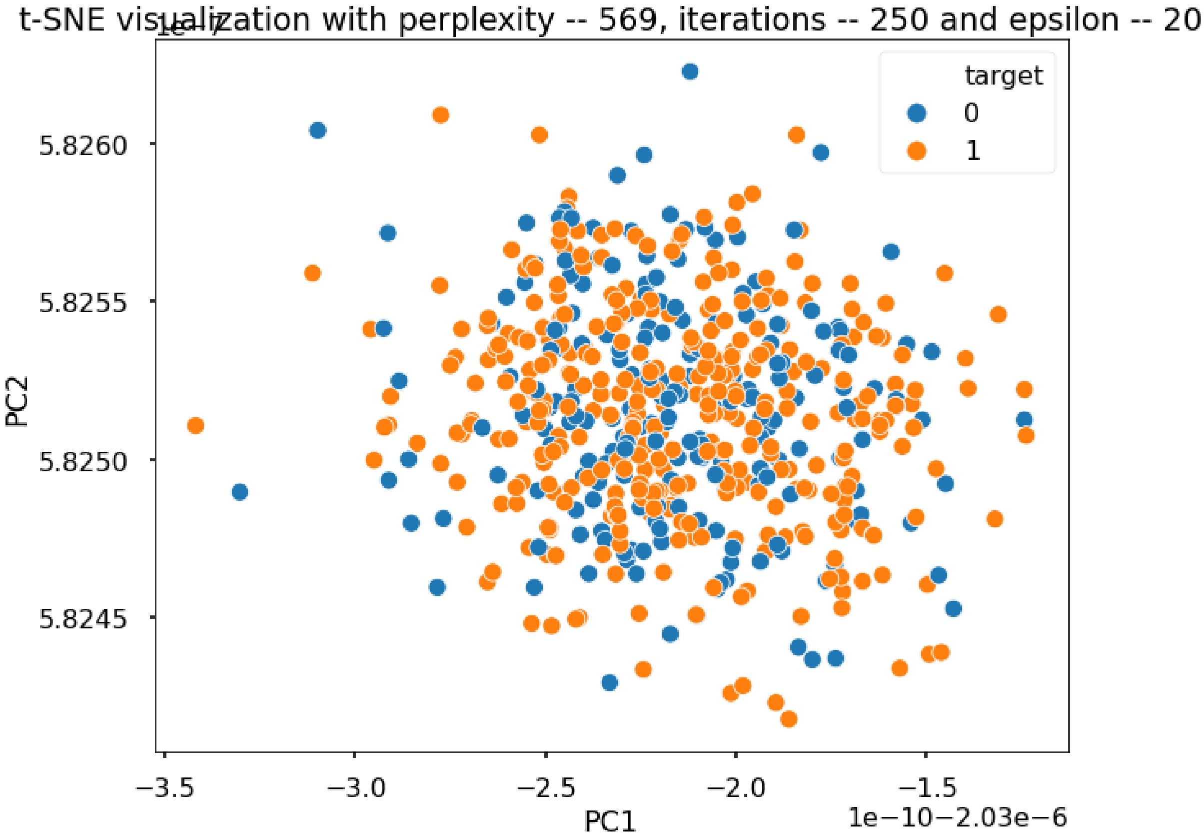


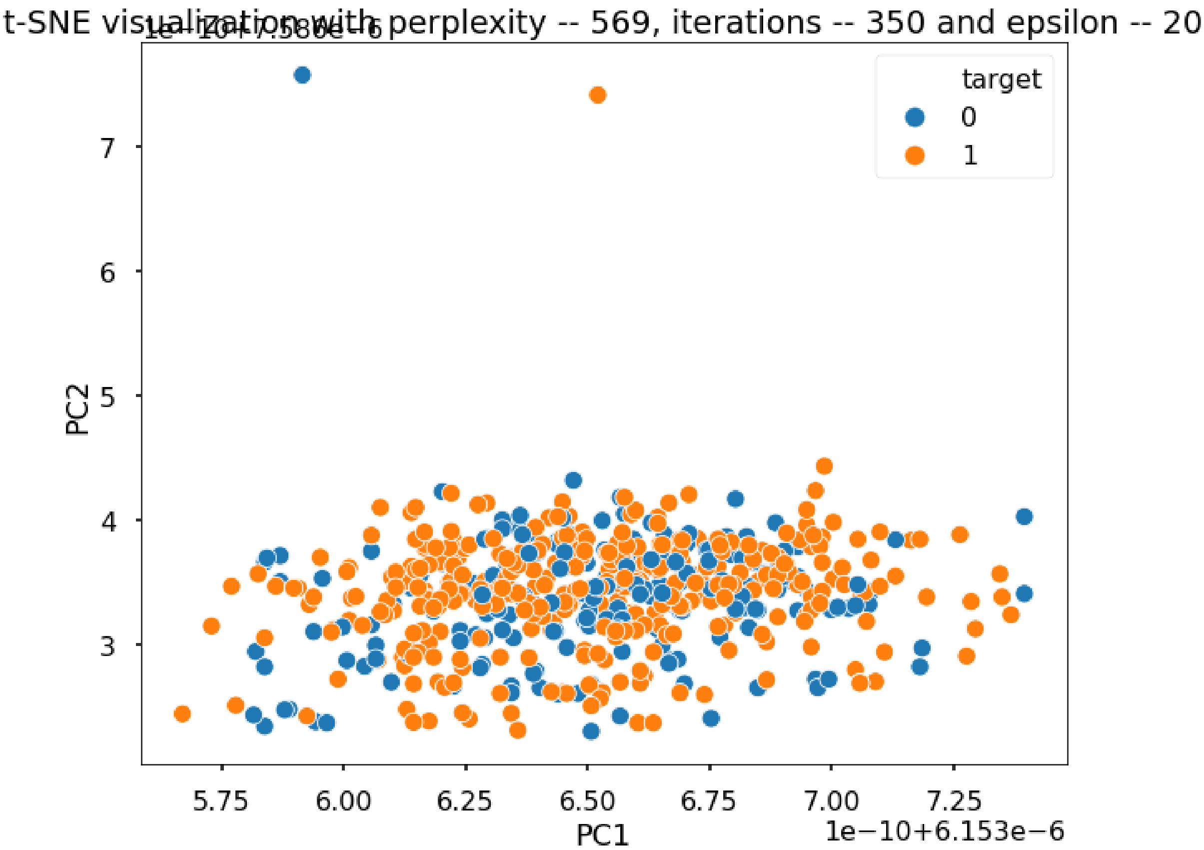
With the lowest value of Perplexity as 1 and Learning rate(Epsilon) slightly on a higher side as 200, we got the small random dense clusters of both the classes.

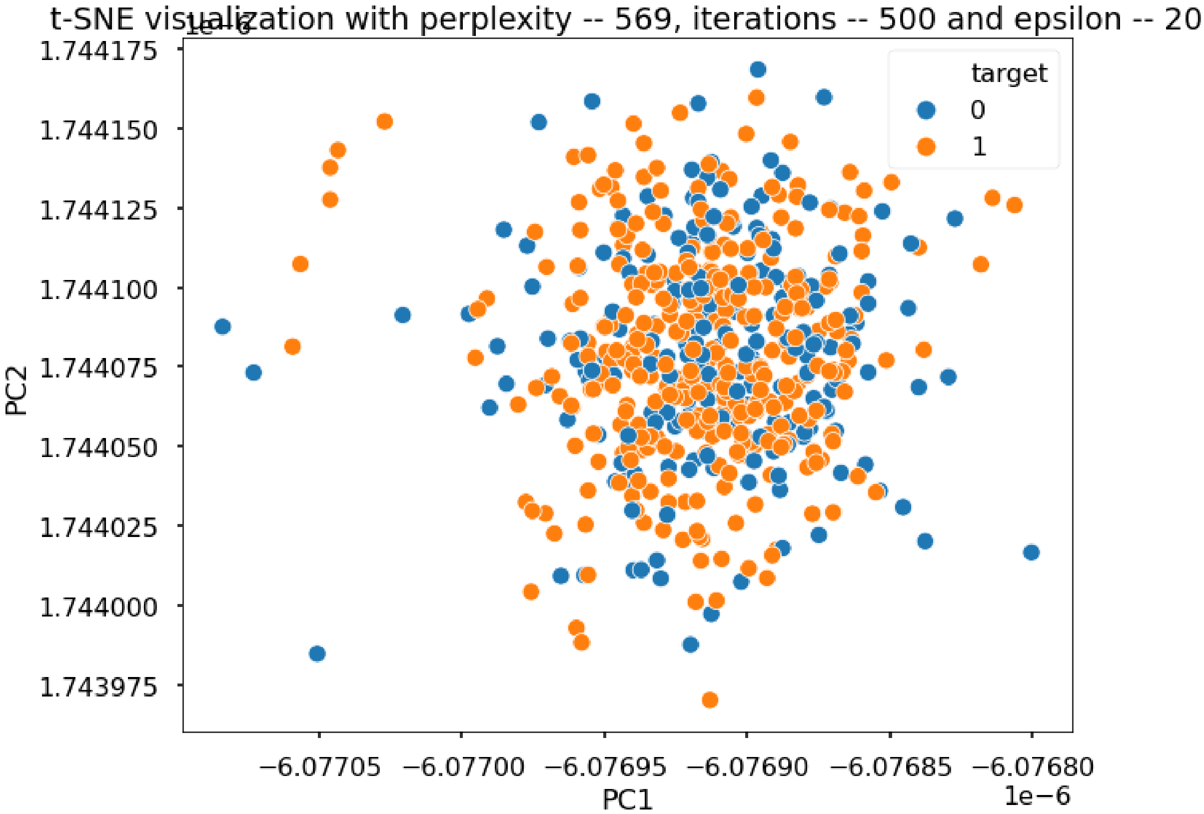
Boundary:CASE-II

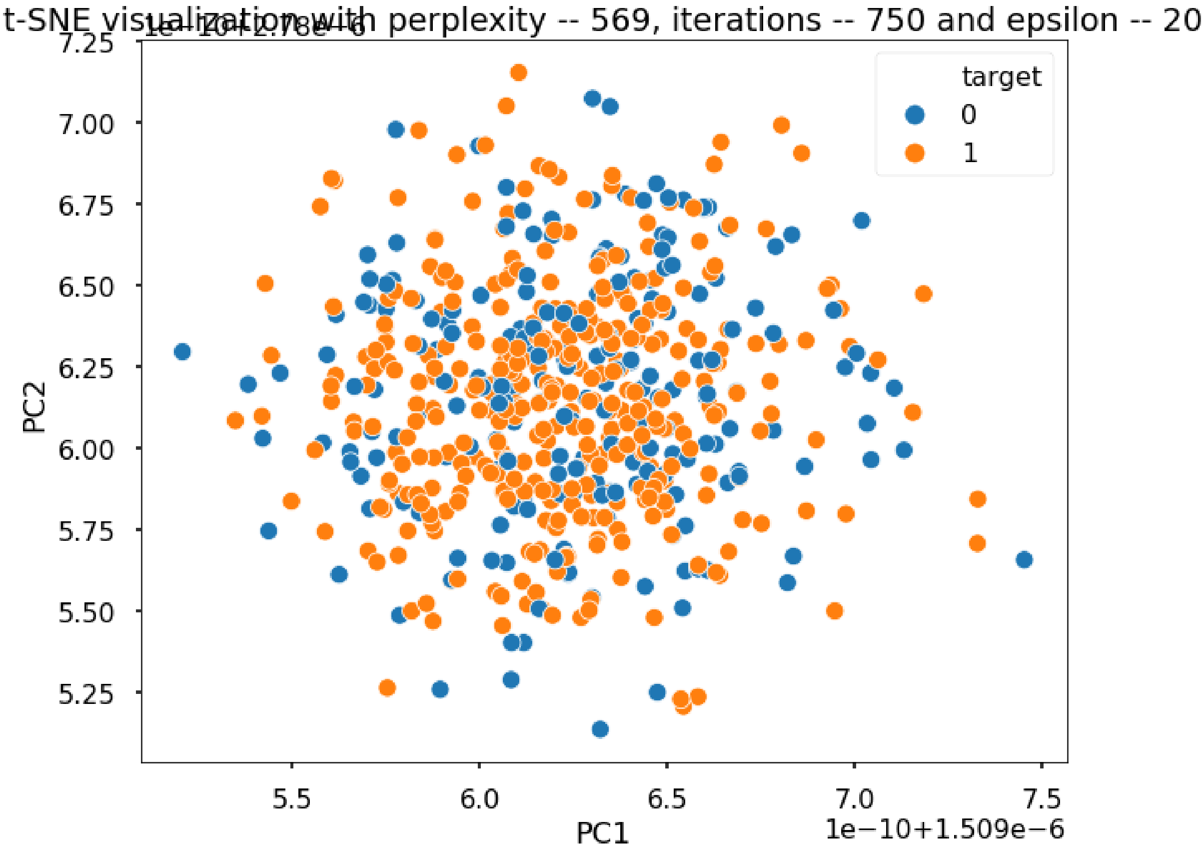
- *### Highest boundary value of Perplexity as 569(number of data points) with Epsilon as low as 20 on different Iterations*

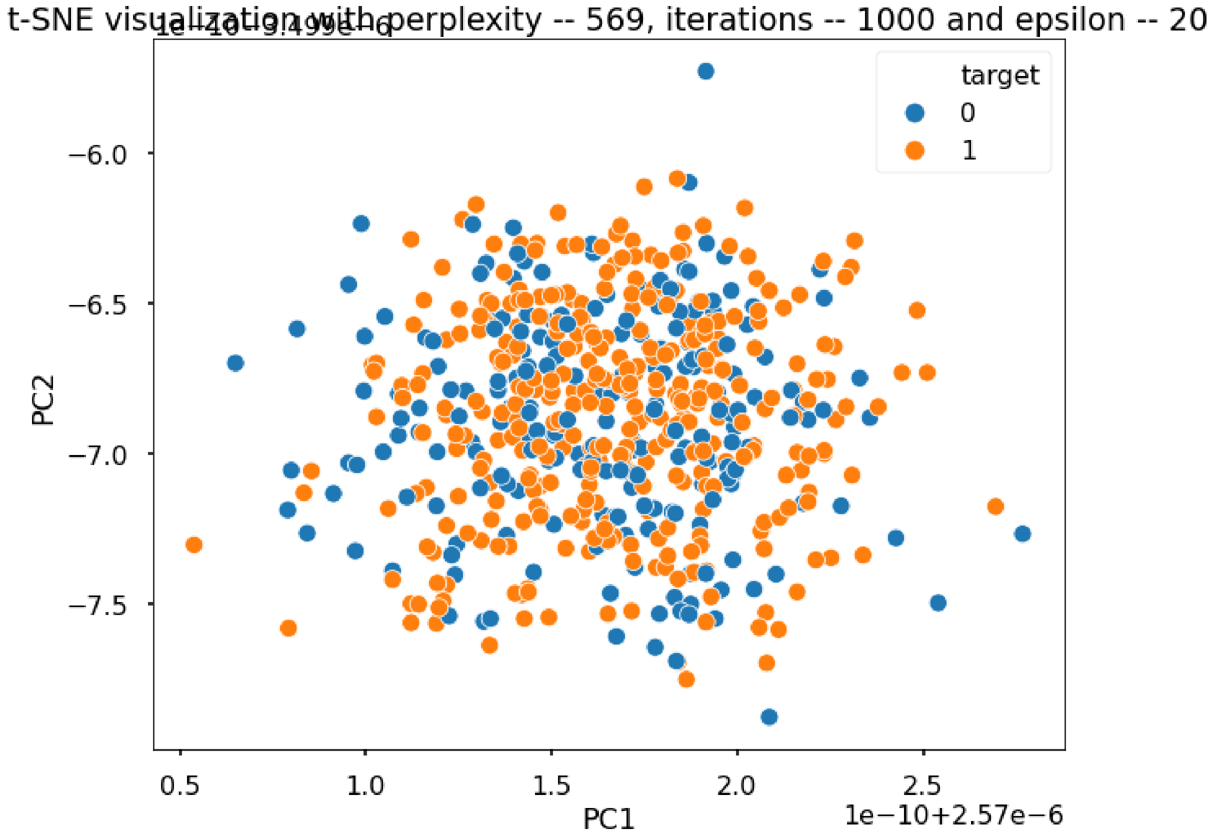
In [5]:
```
iterations = [250,350,500,750,1000,2500,3500,4000,5000]

for idx in range(len(iterations)):
    tsne1 = TSNE(n_components=2,perplexity=569,learning_rate=20,n_iter=iterations[id
    cancer_tsne_pcmps = pd.DataFrame(tsne1.fit_transform(cancer_norm_df),columns=['P
    cancer_tsne_pcmps = pd.concat([cancer_tsne_pcmps,cancer_df['target']],axis=1)
    with plt.style.context('seaborn-poster'):
        plt.figure(figsize=(10,8))
        sns.scatterplot(data=cancer_tsne_pcmps,x='PC1',y='PC2',hue='target')
        plt.title("t-SNE visualization with perplexity -- {0}, iterations -- {1} and


        plt.show()
```

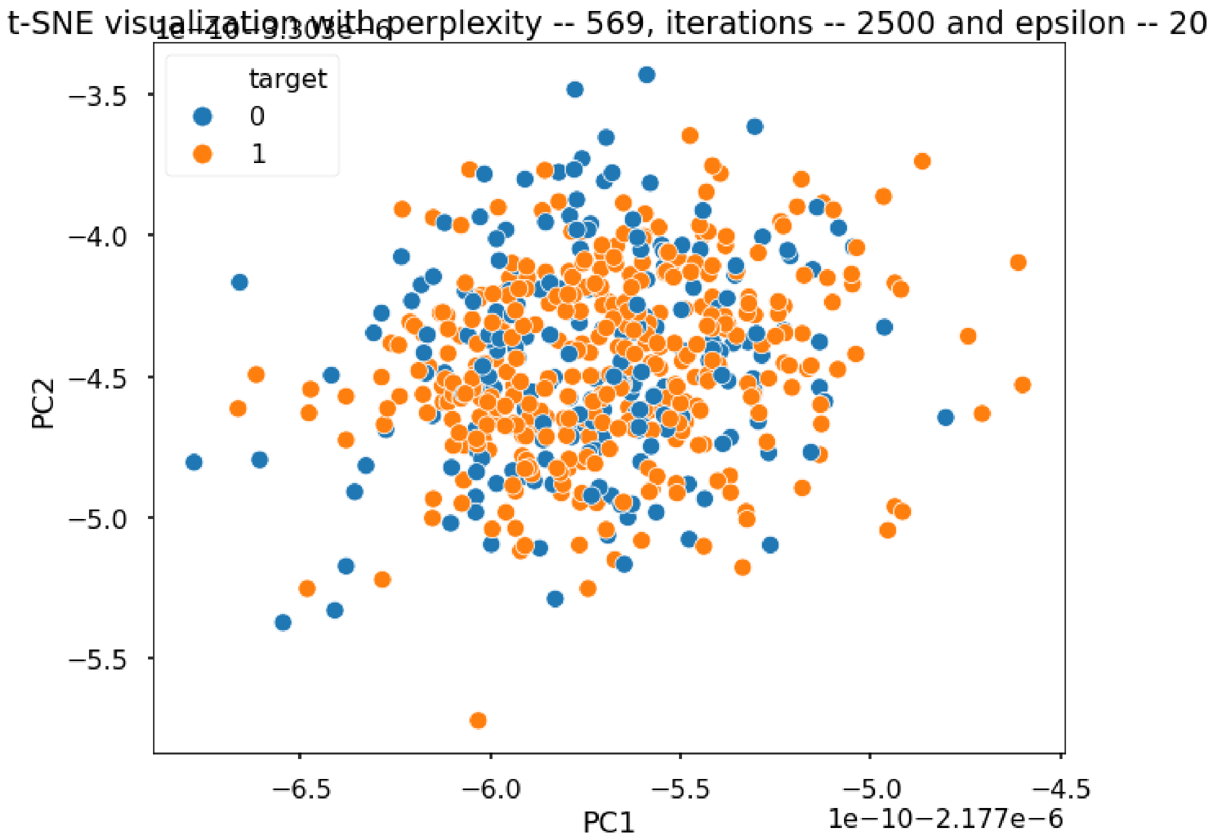

t-SNE visualization with perplexity -- 569, iterations -- 250 and epsilon -- 20
1e−7
5.8260
5.8255
5.8250
5.8245
PC2
target
0
1
−3.5
−3.0
−2.5
−2.0
−1.5
PC1
1e−10−2.03e−6

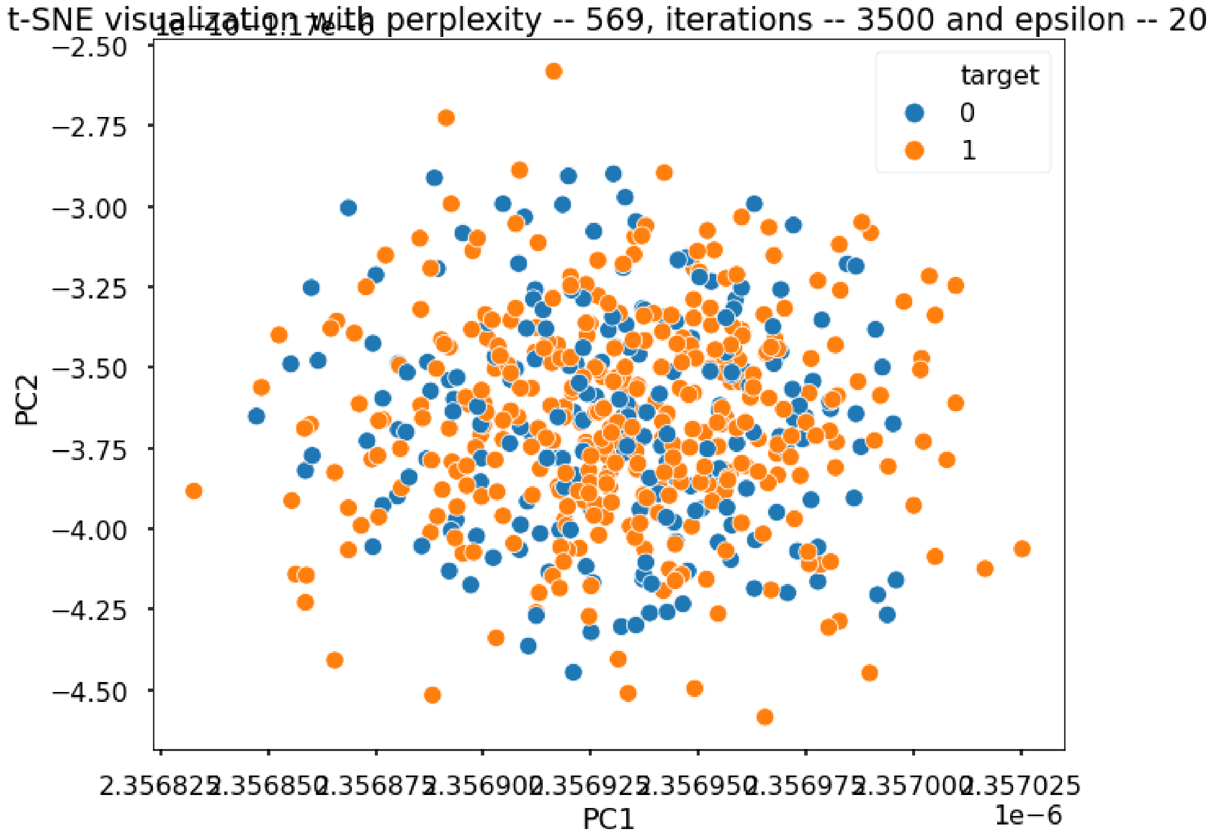

t-SNE visualization with perplexity -- 569, iterations -- 350 and epsilon -- 20
1e−10+7.586e−6
7
6
5
4
3
PC2
target
0
1
5.75
6.00
6.25
6.50
6.75
7.00
7.25
PC1
1e−10+6.153e−6

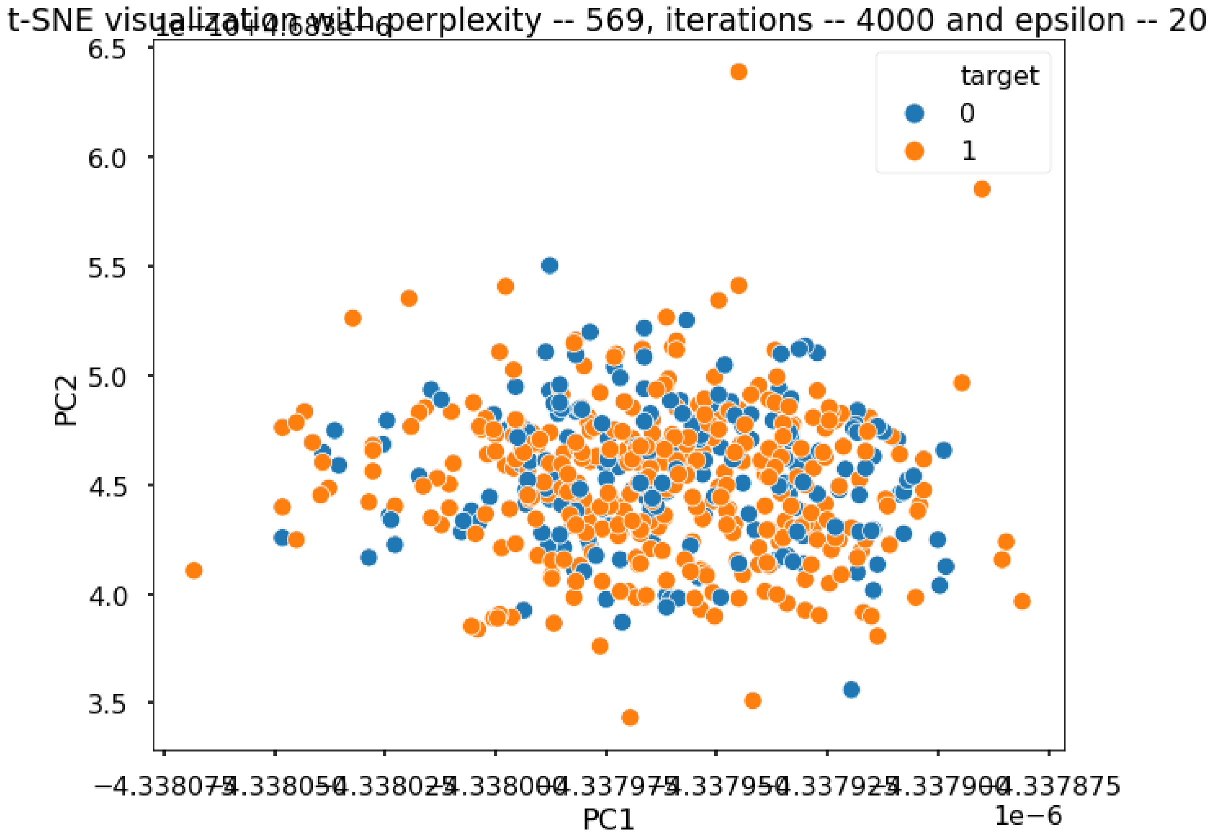

t-SNE visualization with perplexity -- 569, iterations -- 500 and epsilon -- 20
1e−6
1.744175
1.744150
1.744125
1.744100
1.744075
1.744050
1.744025
1.744000
1.743975
PC2
target
0
1
−6.07705
−6.07700
−6.07695
−6.07690
−6.07685
−6.07680
1e−6
PC1

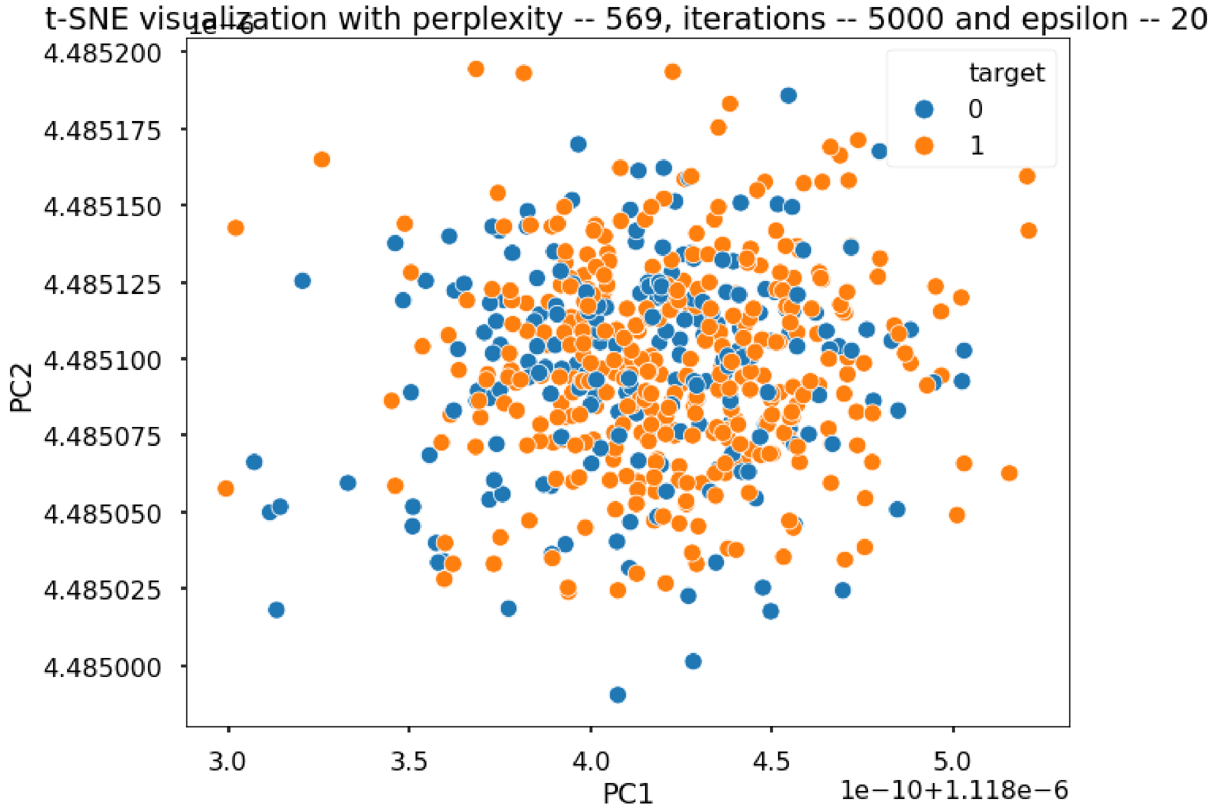

t-SNE visualization with perplexity -- 569, iterations -- 750 and epsilon -- 20
1e−10+2.78e−6
7.25
7.00
6.75
6.50
6.25
6.00
5.75
5.50
5.25
PC2
target
0
1
5.5
6.0
6.5
7.0
7.5
1e−10+1.509e−6
PC1


t-SNE visualization with perplexity -- 569, iterations -- 1000 and epsilon -- 20
target
0
1
−6.0
−6.5
−7.0
−7.5
PC2
0.5
1.0
1.5
2.0
2.5
PC1
1e−10+2.57e−6


t-SNE visualization with perplexity -- 569, iterations -- 2500 and epsilon -- 20
target
0
1
−3.5
−4.0
−4.5
−5.0
−5.5
PC2
−6.5
−6.0
−5.5
−5.0
−4.5
PC1
1e−10−2.177e−6


t-SNE visualization with perplexity -- 569, iterations -- 3500 and epsilon -- 20
1e−10+1.17e−6
target
0
1
PC2
PC1
1e−6
−2.50
−2.75
−3.00
−3.25
−3.50
−3.75
−4.00
−4.25
−4.50
2.356825
2.356850
2.356875
2.356900
2.356925
2.356950
2.356975
2.357000
2.357025


t-SNE visualization with perplexity -- 569, iterations -- 4000 and epsilon -- 20
1e−10+4.683e−6
target
0
1
PC2
PC1
1e−6
6.5
6.0
5.5
5.0
4.5
4.0
3.5
−4.338075
−4.338050
−4.338025
−4.338000
−4.337975
−4.337950
−4.337925
−4.337900
−4.337875



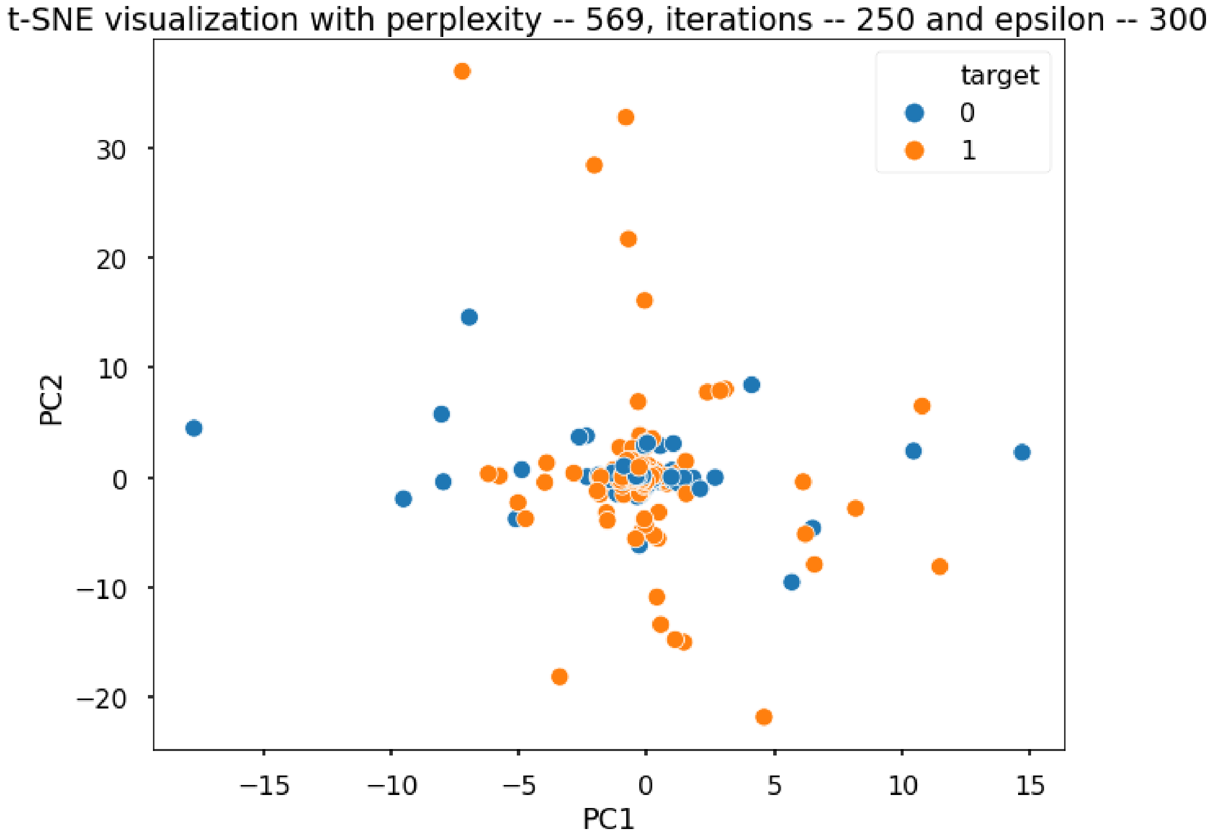
With the highest value of Perplexity as 569 and Learning rate(Epsilon) on a lower side as 20, we got the completely random data as a Noise.

Boundary:CASE-III

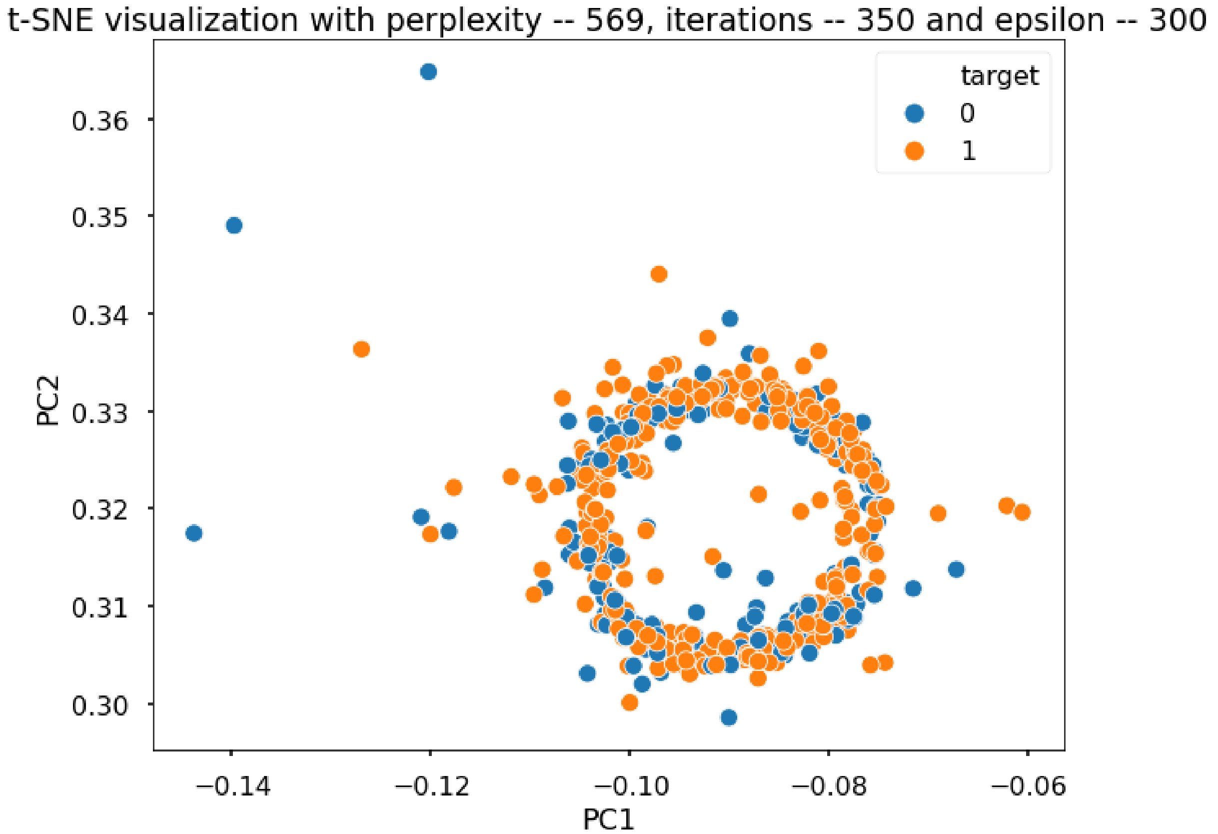
- *### Highest boundary value of Perplexity as 569(number of data points) with Epsilon as high as 300 on different Iterations*

In [6]:

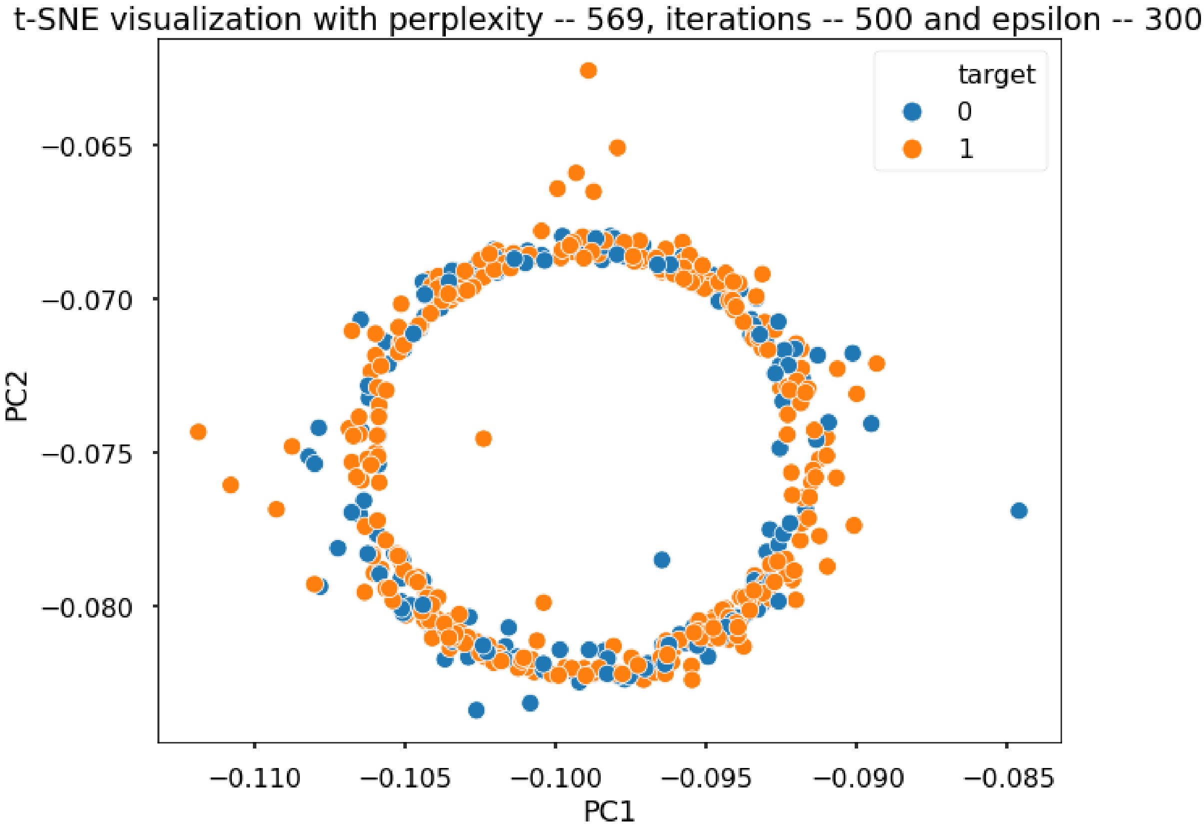
```
iterations = [250,350,500,750,1000,2500,3500,4000,5000]

for idx in range(len(iterations)):
    tsne1 = TSNE(n_components=2,perplexity=569,learning_rate=300,n_iter=iterations[i
    cancer_tsne_pcmps = pd.DataFrame(tsne1.fit_transform(cancer_norm_df),columns=['P
    cancer_tsne_pcmps = pd.concat([cancer_tsne_pcmps,cancer_df['target']],axis=1)
    with plt.style.context('seaborn-poster'):
        plt.figure(figsize=(10,8))
        sns.scatterplot(data=cancer_tsne_pcmps,x='PC1',y='PC2',hue='target')
        plt.title("t-SNE visualization with perplexity -- {0}, iterations -- {1} and


        plt.show()
```

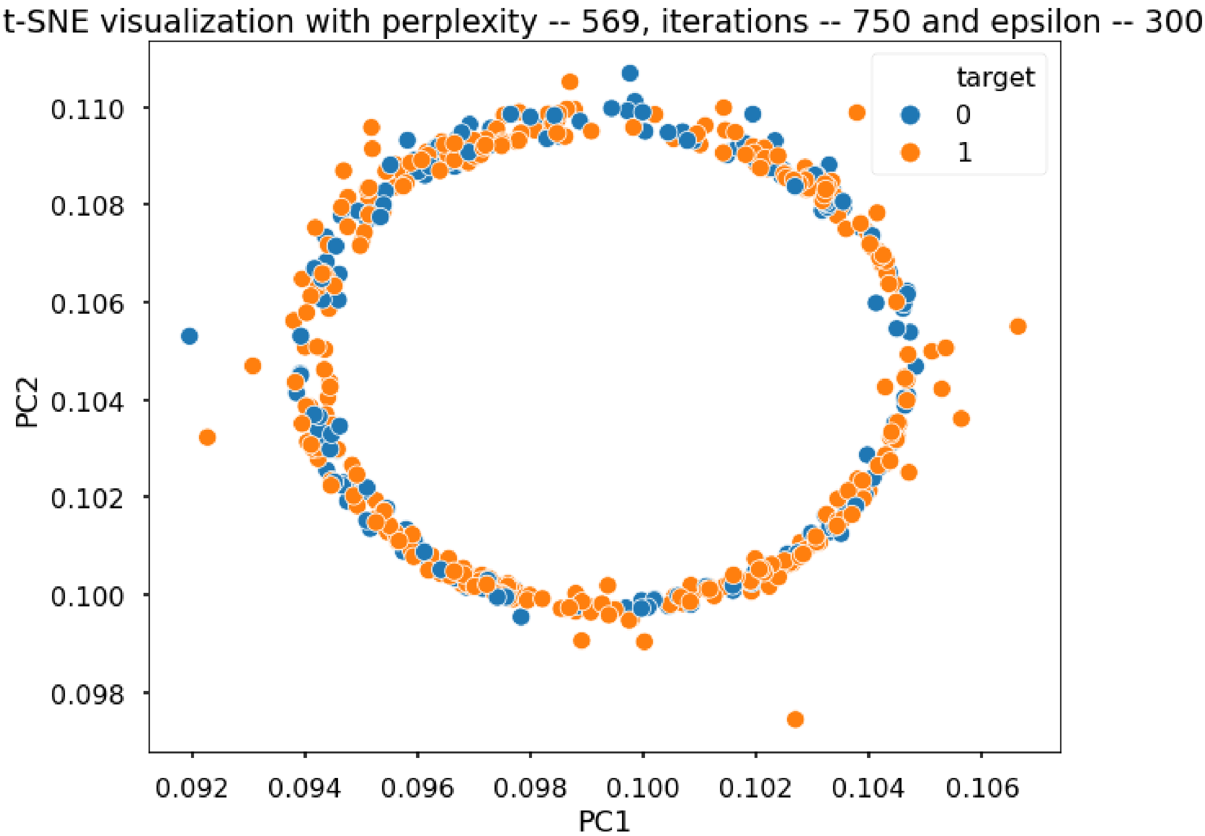


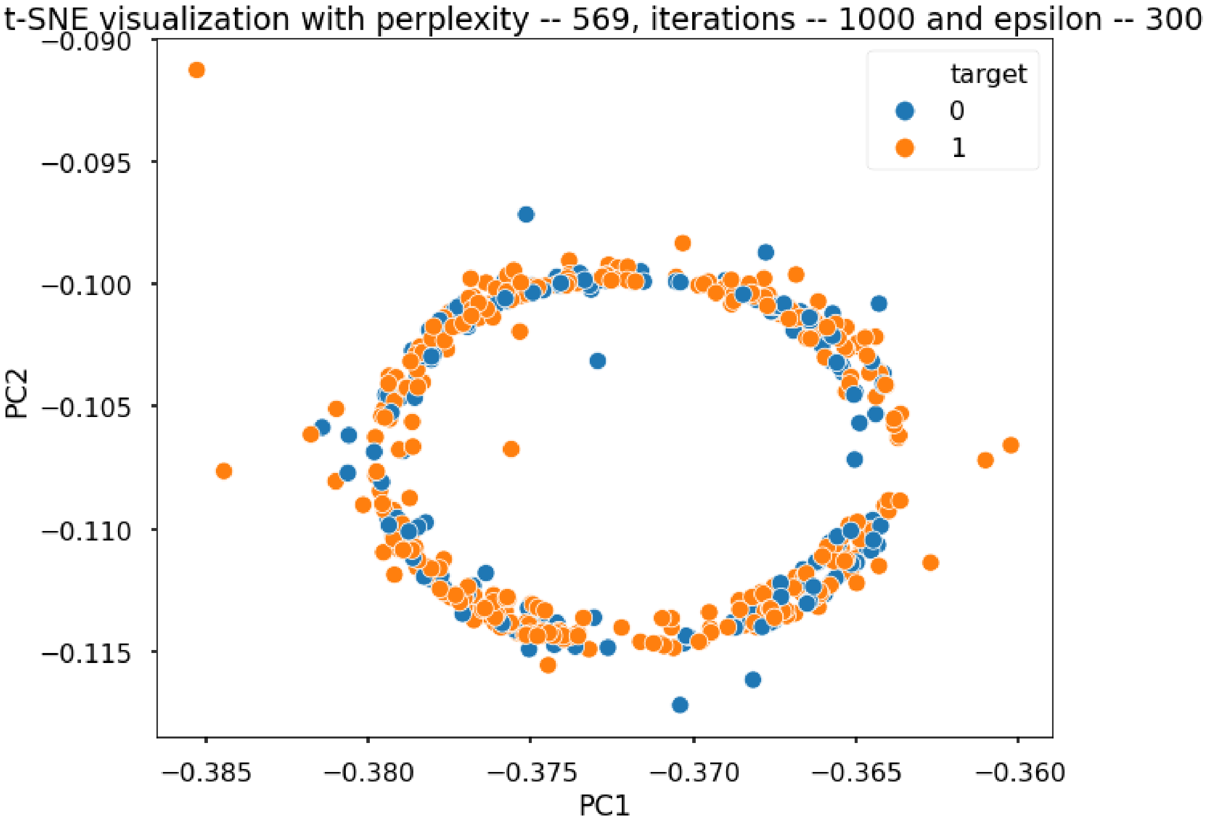
t-SNE visualization with perplexity -- 569, iterations -- 250 and epsilon -- 300
target
0
1
PC2
PC1
30
20
10
0
−10
−20
−15
−10
−5
0
5
10
15

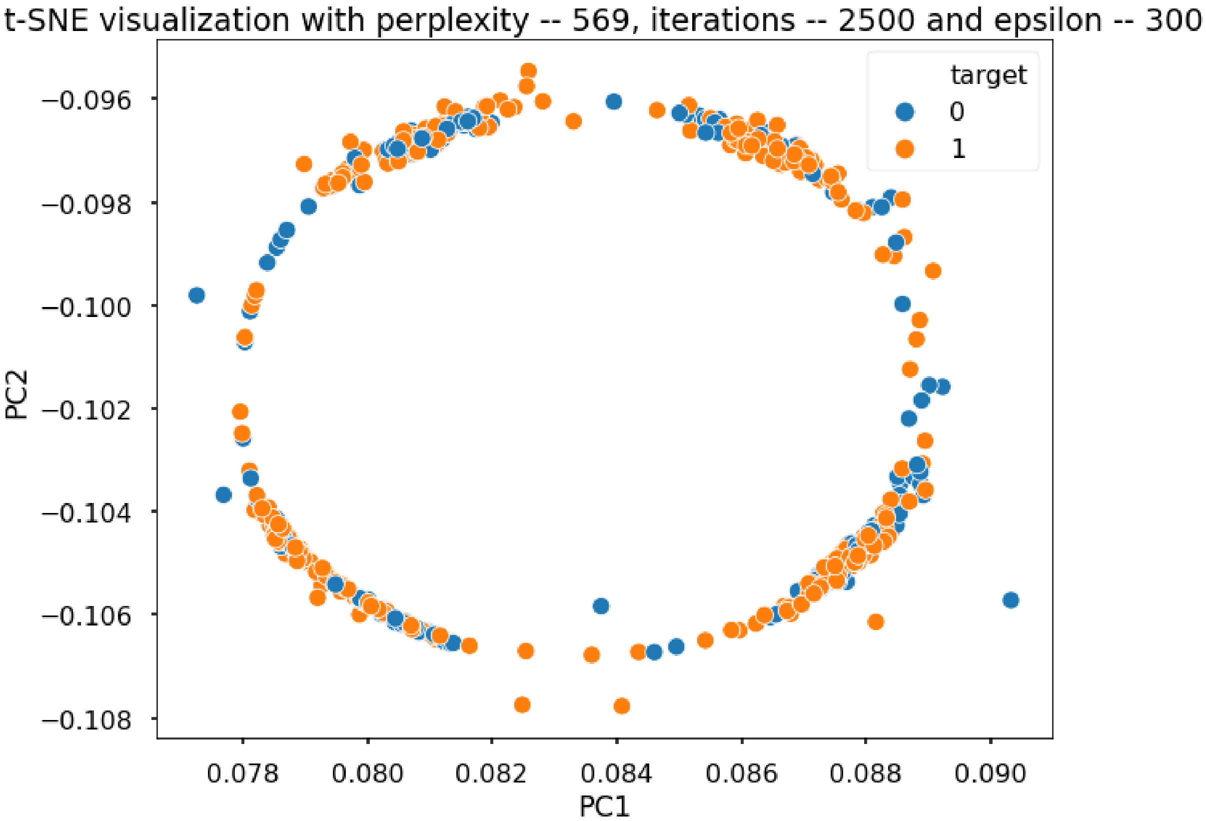


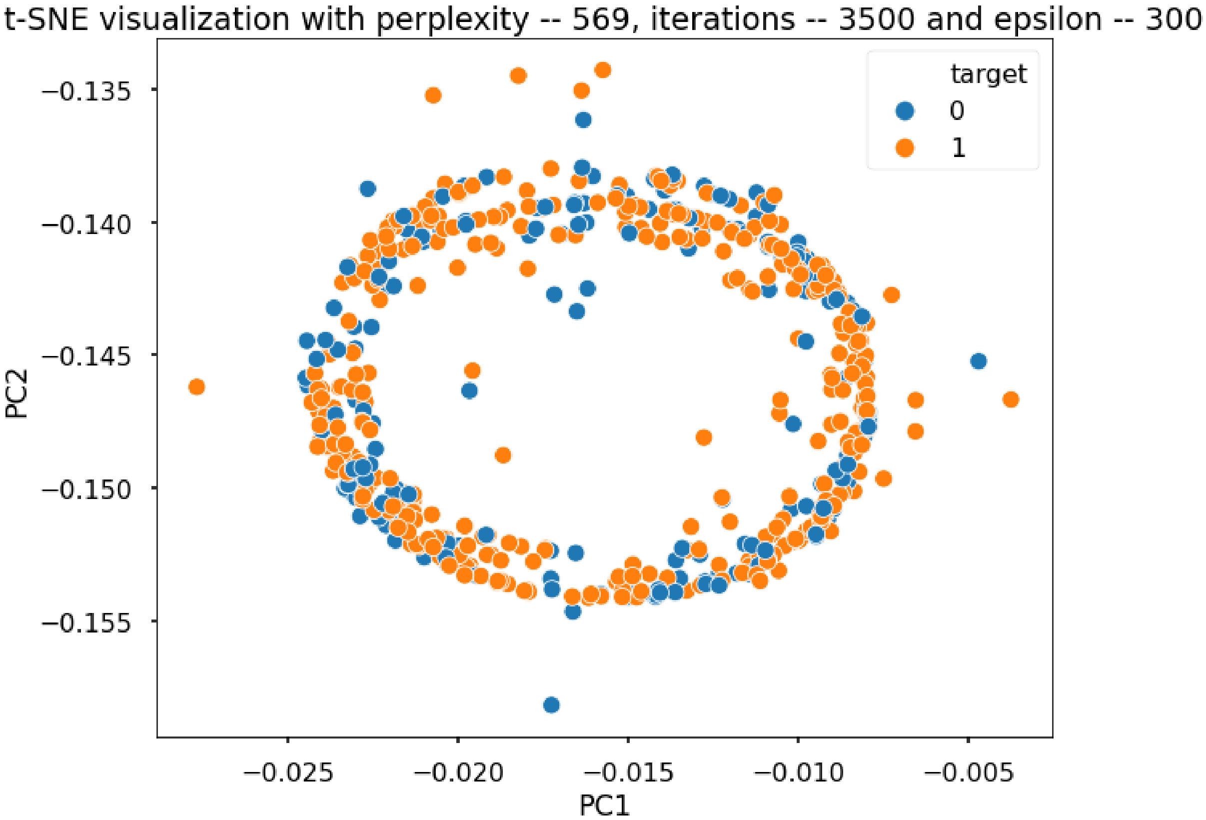
t-SNE visualization with perplexity -- 569, iterations -- 350 and epsilon -- 300
target
0
1
PC2
PC1
0.36
0.35
0.34
0.33
0.32
0.31
0.30
−0.14
−0.12
−0.10
−0.08
−0.06

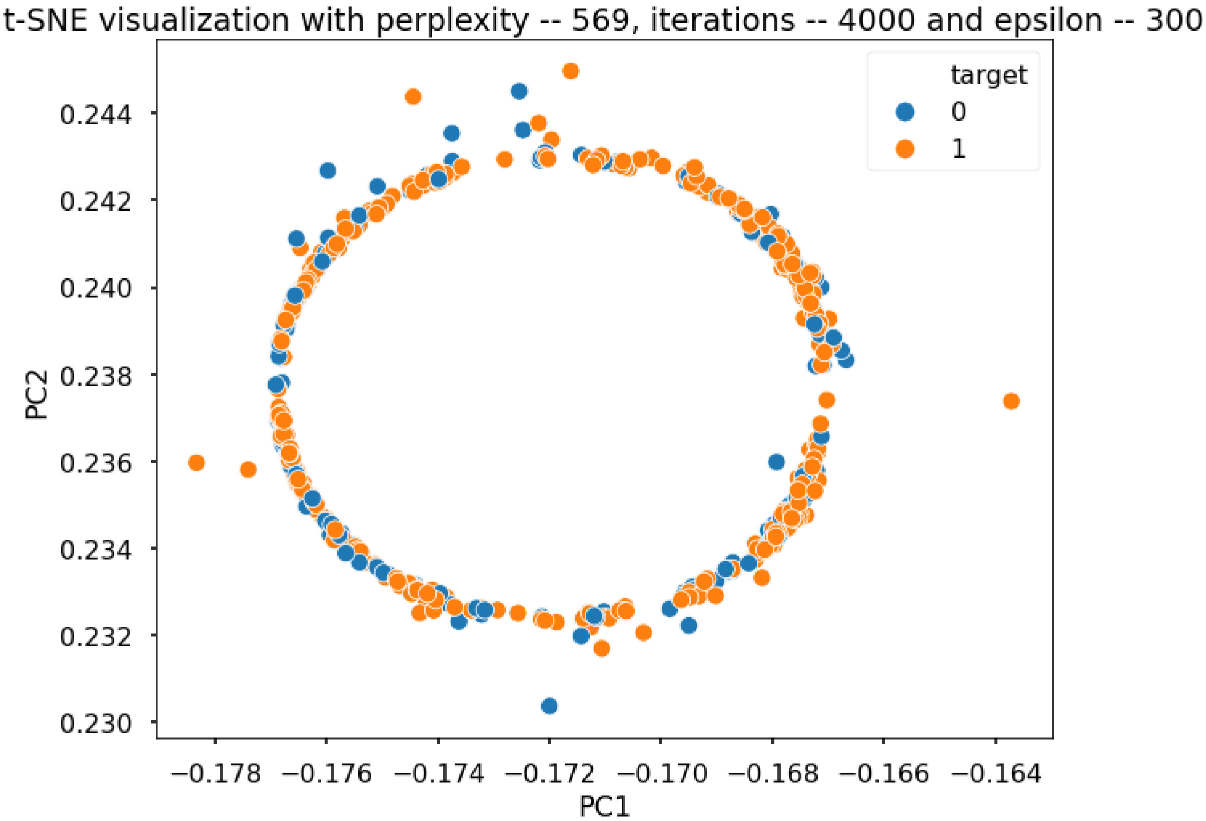

t-SNE visualization with perplexity -- 569, iterations -- 500 and epsilon -- 300
target
0
1
−0.065
−0.070
−0.075
−0.080
PC2
−0.110
−0.105
−0.100
−0.095
−0.090
−0.085
PC1

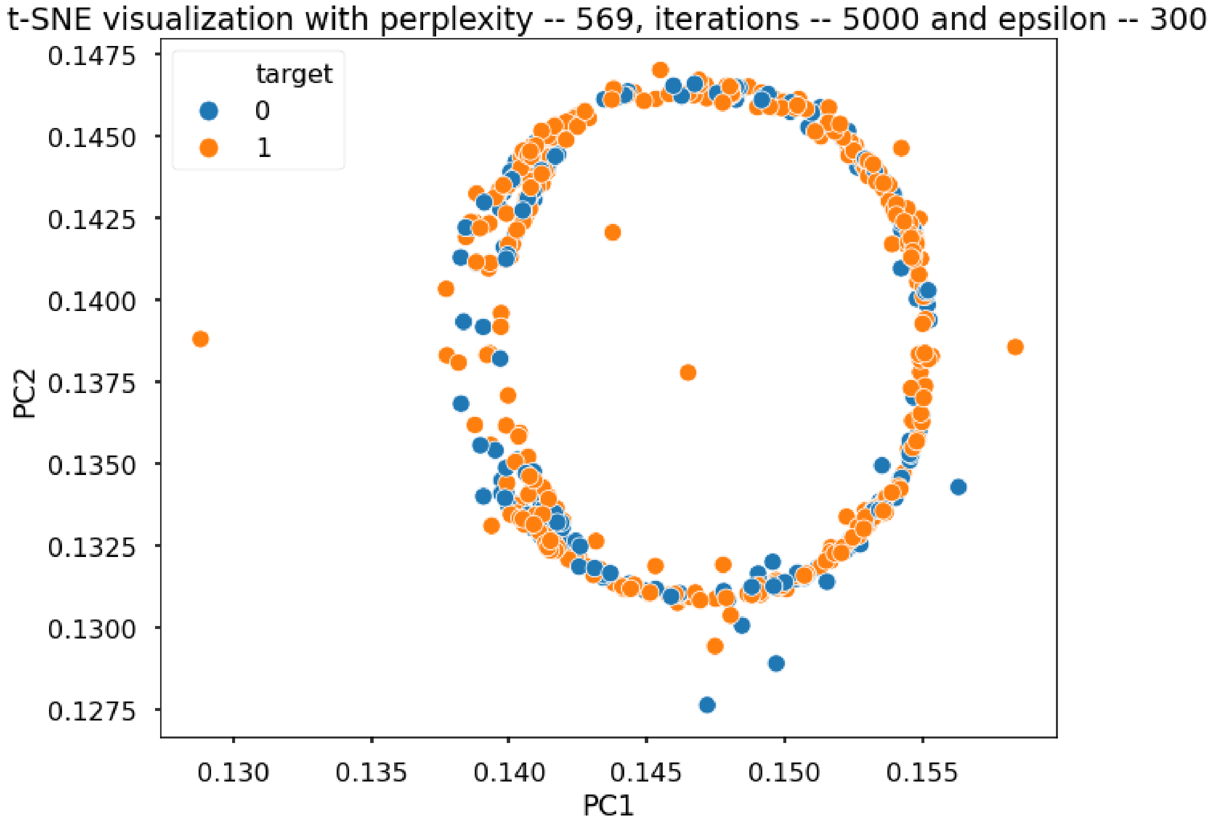

t-SNE visualization with perplexity -- 569, iterations -- 750 and epsilon -- 300
target
0
1
0.110
0.108
0.106
0.104
0.102
0.100
0.098
PC2
0.092
0.094
0.096
0.098
0.100
0.102
0.104
0.106
PC1


t-SNE visualization with perplexity -- 569, iterations -- 1000 and epsilon -- 300
target
0
1
PC2
PC1
−0.090
−0.095
−0.100
−0.105
−0.110
−0.115
−0.385
−0.380
−0.375
−0.370
−0.365
−0.360


t-SNE visualization with perplexity -- 569, iterations -- 2500 and epsilon -- 300
target
0
1
PC2
PC1
−0.096
−0.098
−0.100
−0.102
−0.104
−0.106
−0.108
0.078
0.080
0.082
0.084
0.086
0.088
0.090


t-SNE visualization with perplexity -- 569, iterations -- 3500 and epsilon -- 300
target
0
1
PC2
PC1
−0.135
−0.140
−0.145
−0.150
−0.155
−0.025
−0.020
−0.015
−0.010
−0.005


t-SNE visualization with perplexity -- 569, iterations -- 4000 and epsilon -- 300
target
0
1
PC2
PC1
0.244
0.242
0.240
0.238
0.236
0.234
0.232
0.230
−0.178
−0.176
−0.174
−0.172
−0.170
−0.168
−0.166
−0.164



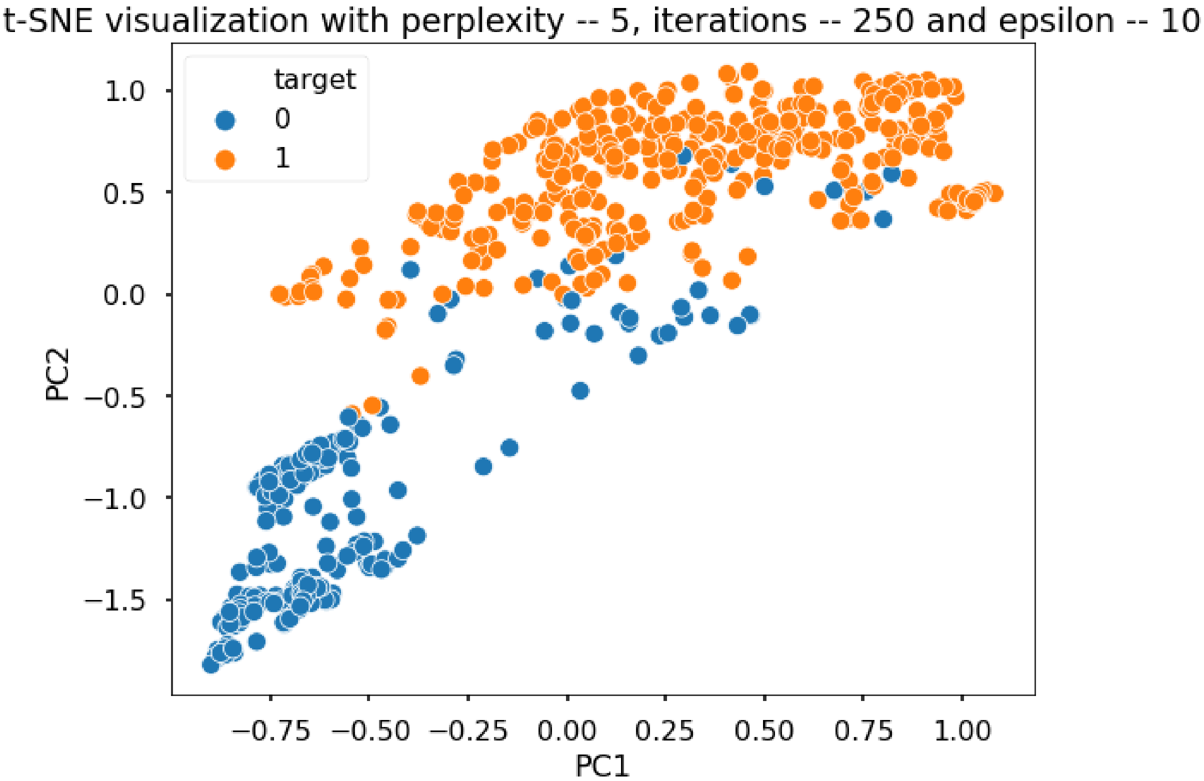
With the highest value of Perplexity as 569 and Learning rate(Epsilon) on a higher side as 300, we got the ball shaped data cluster which says that every point is a definite distance away from each other.

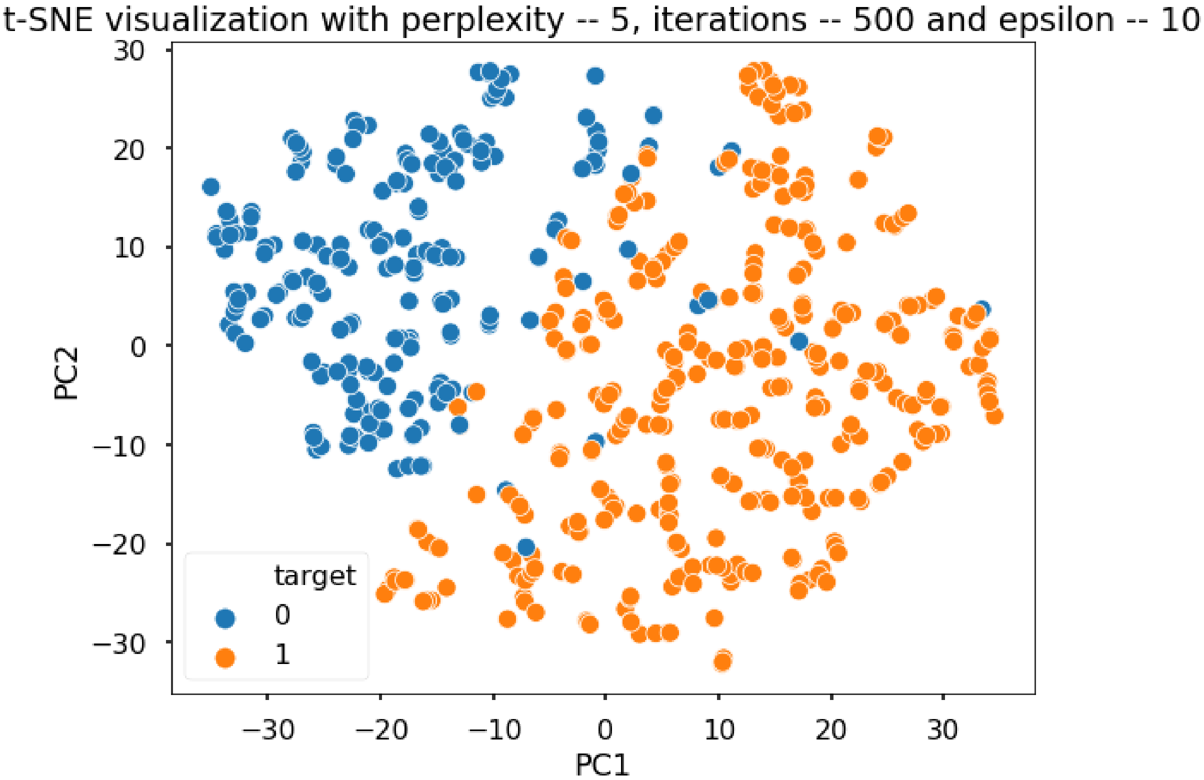
Multiple_runs

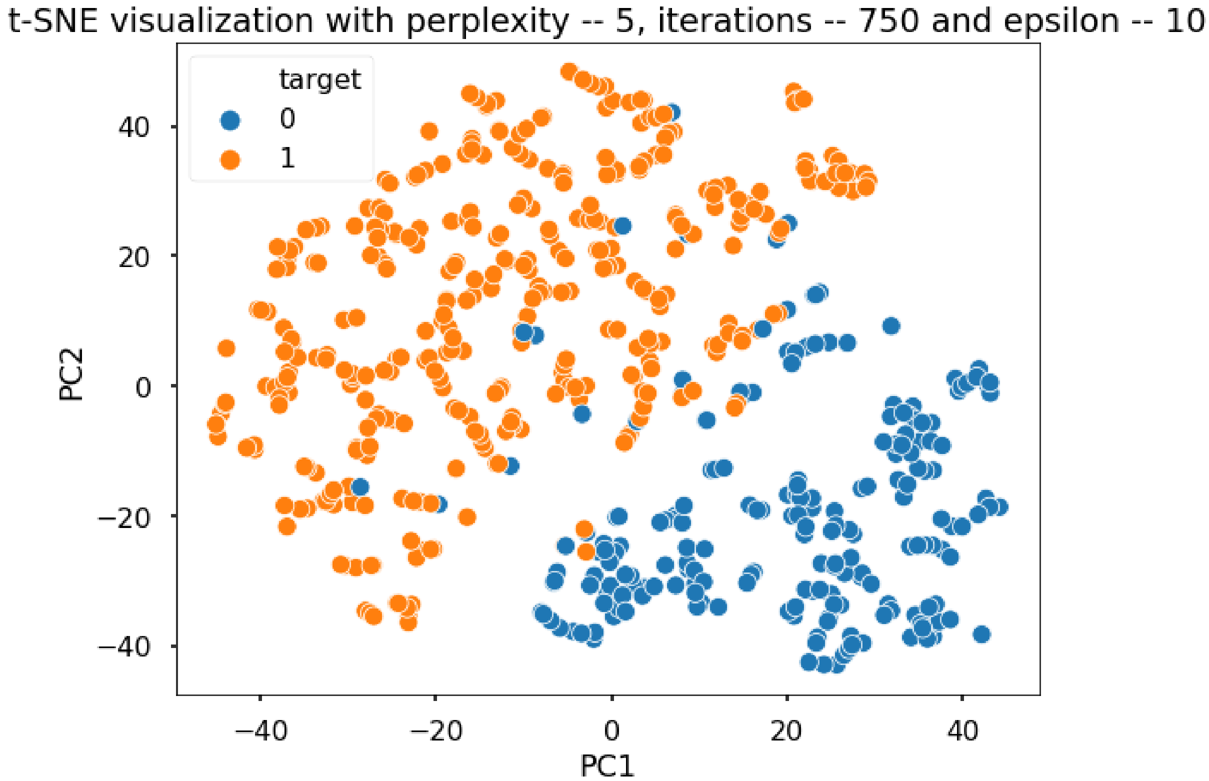
CASE-I-Multiple_runs

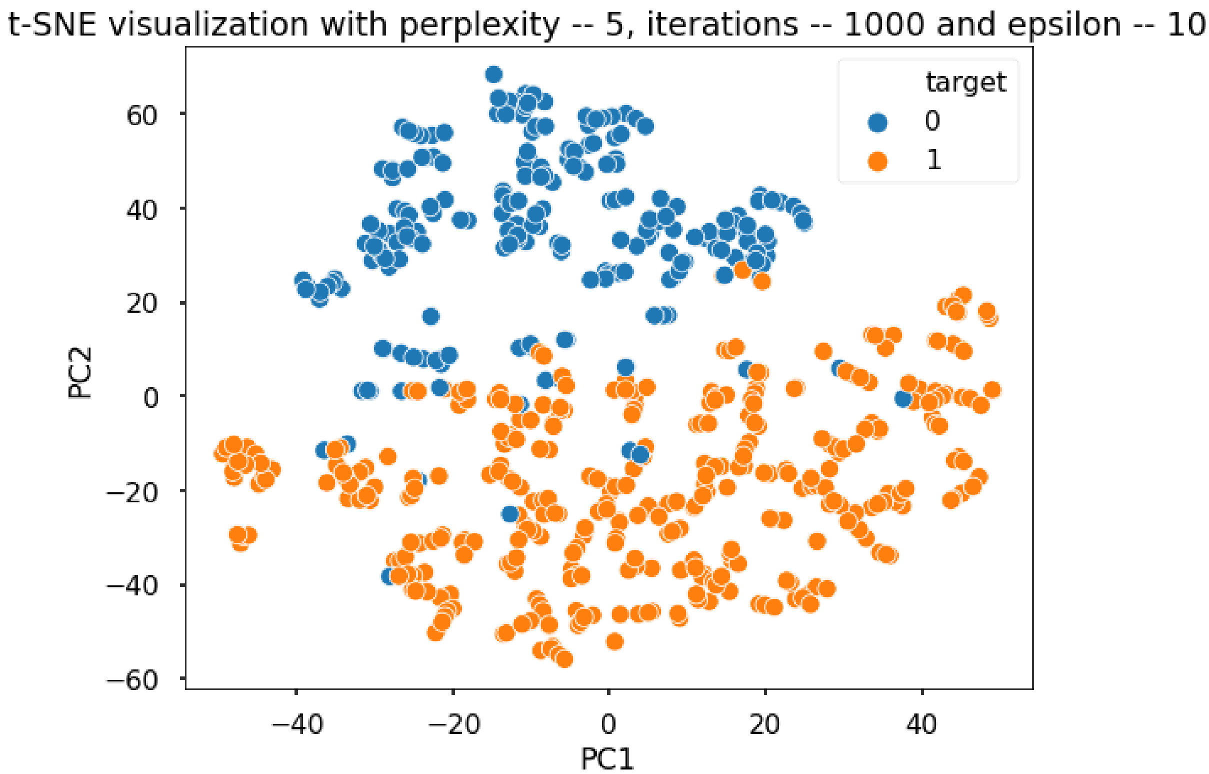
Running t-SNE on a range of perplexities, iterations and learning rates with embedding initialization as 'random'

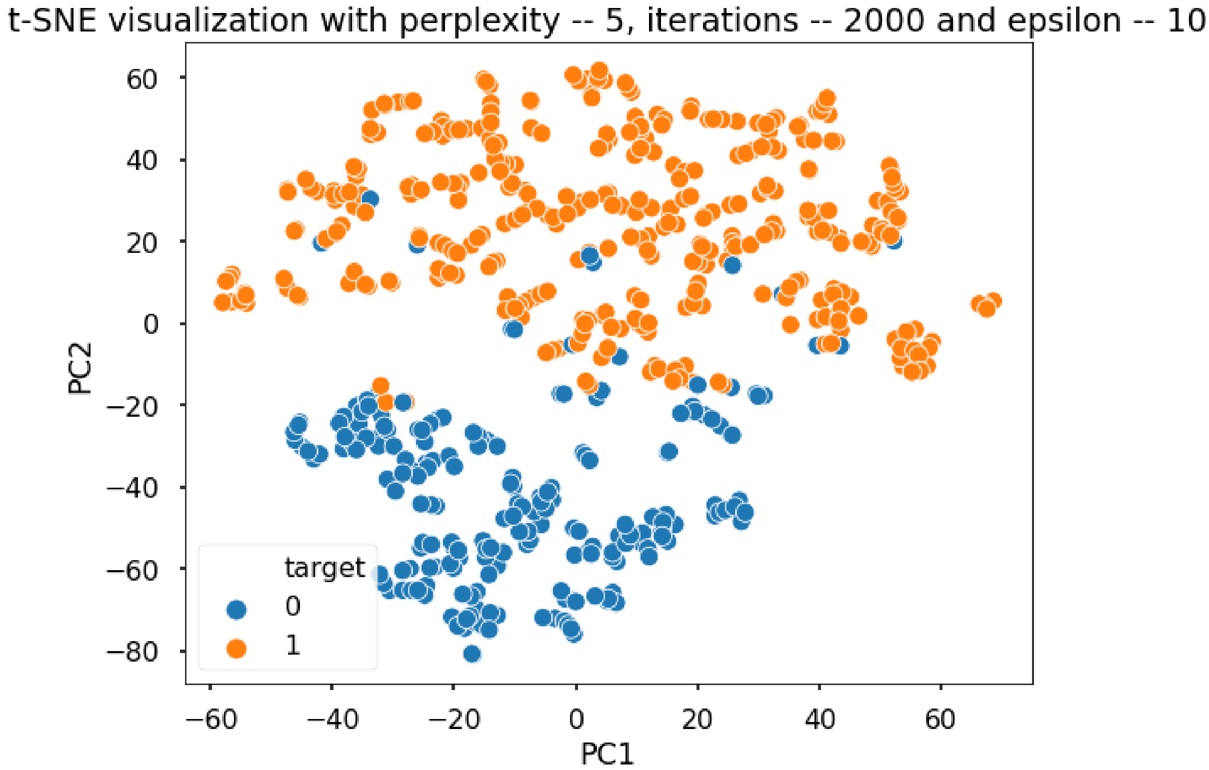
In [8]:
```
perplexities = [30,50,100]
iterations = [250,500,750,1000,3000,5000]
l_rate = [10,30,50,100,200]

for pidx in range(len(perplexities)):
    for lidx in range(len(l_rate)):
        for idx in range(len(iterations)):
            tsne1 = TSNE(n_components=2,init='random',perplexity=perplexities[pidx],
                        n_jobs=-1)
            cancer_tsne_pcmps = pd.DataFrame(tsne1.fit_transform(cancer_norm_df),col
            cancer_tsne_pcmps = pd.concat([cancer_tsne_pcmps,cancer_df['target']],ax
            with plt.style.context('seaborn-poster'):
                plt.figure(figsize=(9,7))
                sns.scatterplot(data=cancer_tsne_pcmps,x='PC1',y='PC2',hue='target')
                plt.title("t-SNE visualization with perplexity -- {0}, iterations --


                plt.show()
```

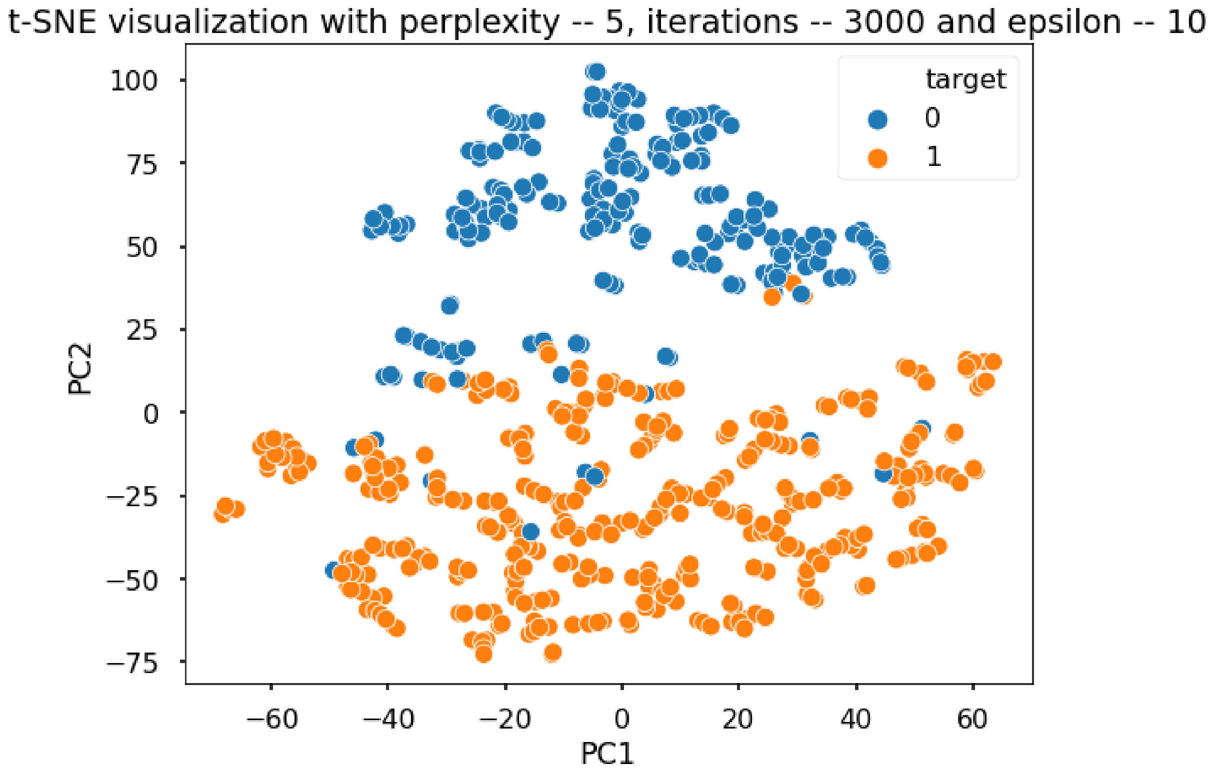

t-SNE visualization with perplexity -- 5, iterations -- 250 and epsilon -- 10
target
0
1
1.0
0.5
0.0
−0.5
−1.0
−1.5
PC2
−0.75
−0.50
−0.25
0.00
0.25
0.50
0.75
1.00
PC1

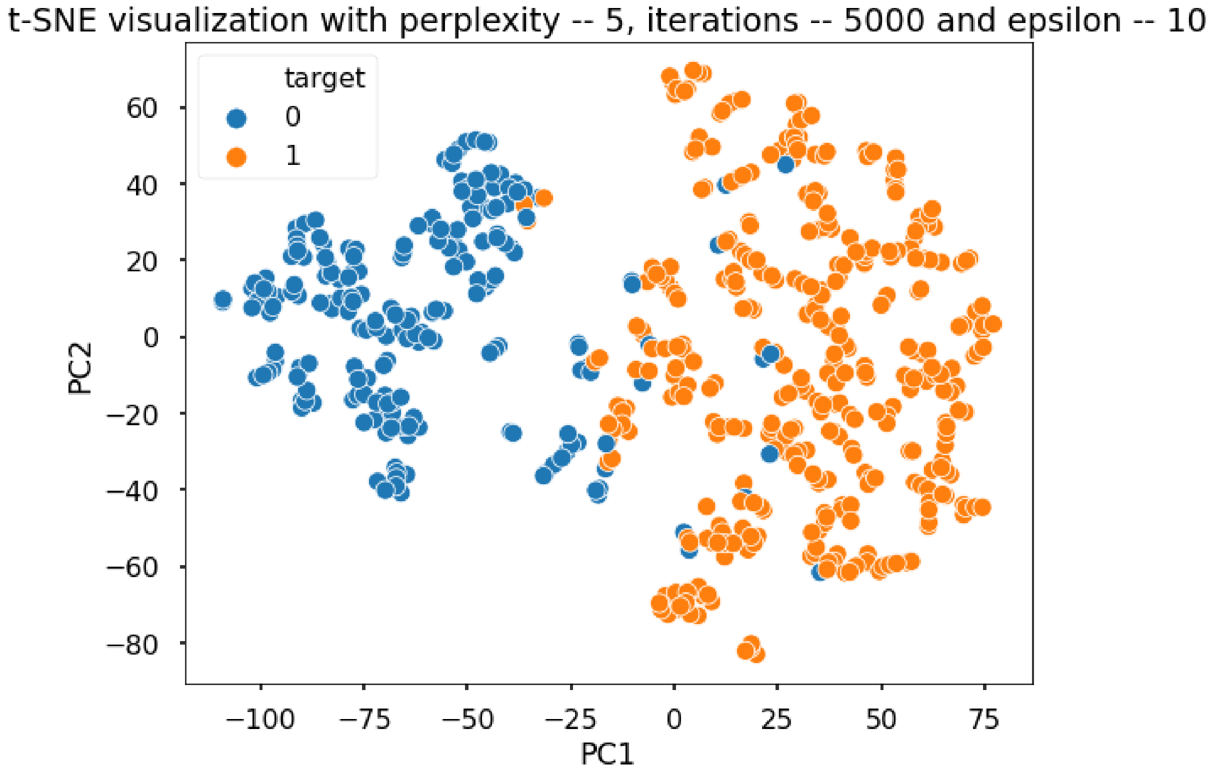

t-SNE visualization with perplexity -- 5, iterations -- 500 and epsilon -- 10
30
20
10
0
−10
−20
−30
PC2
target
0
1
−30
−20
−10
0
10
20
30
PC1

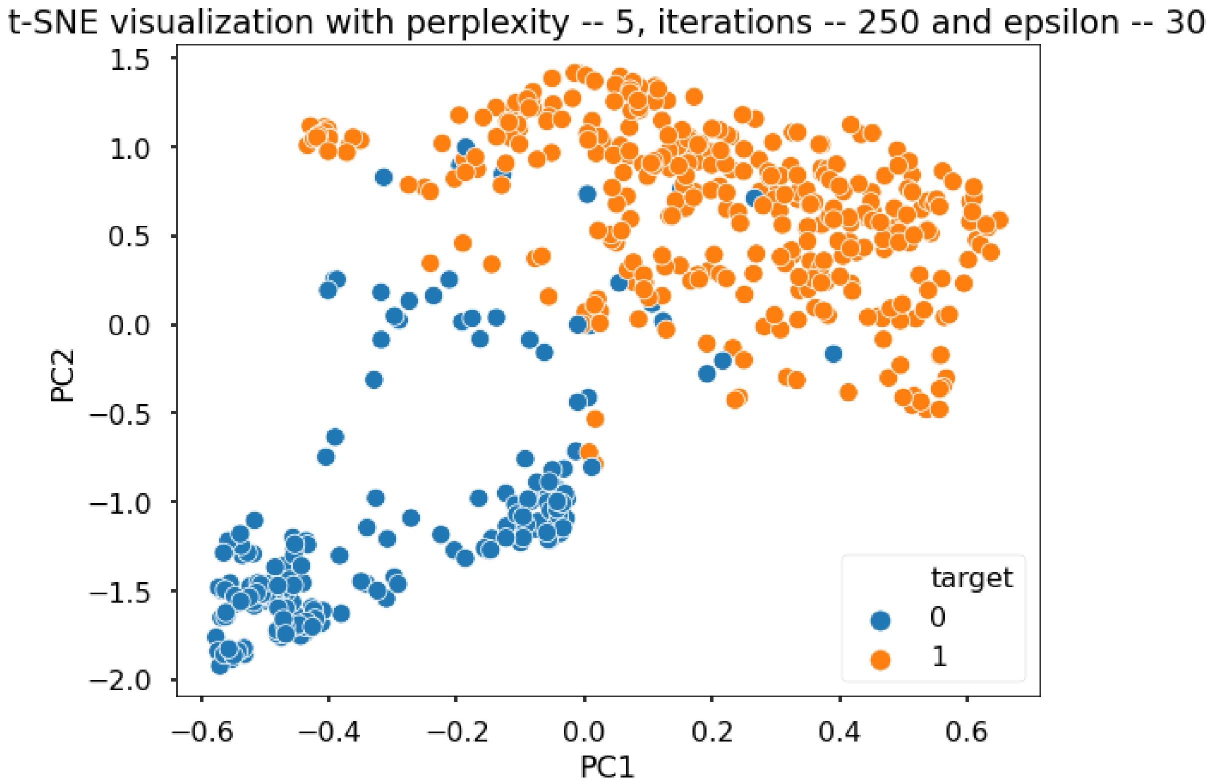

t-SNE visualization with perplexity -- 5, iterations -- 750 and epsilon -- 10
target
0
1
PC2
40
20
0
−20
−40
PC1
−40
−20
0
20
40


t-SNE visualization with perplexity -- 5, iterations -- 1000 and epsilon -- 10
target
0
1
PC2
60
40
20
0
−20
−40
−60
PC1
−40
−20
0
20
40

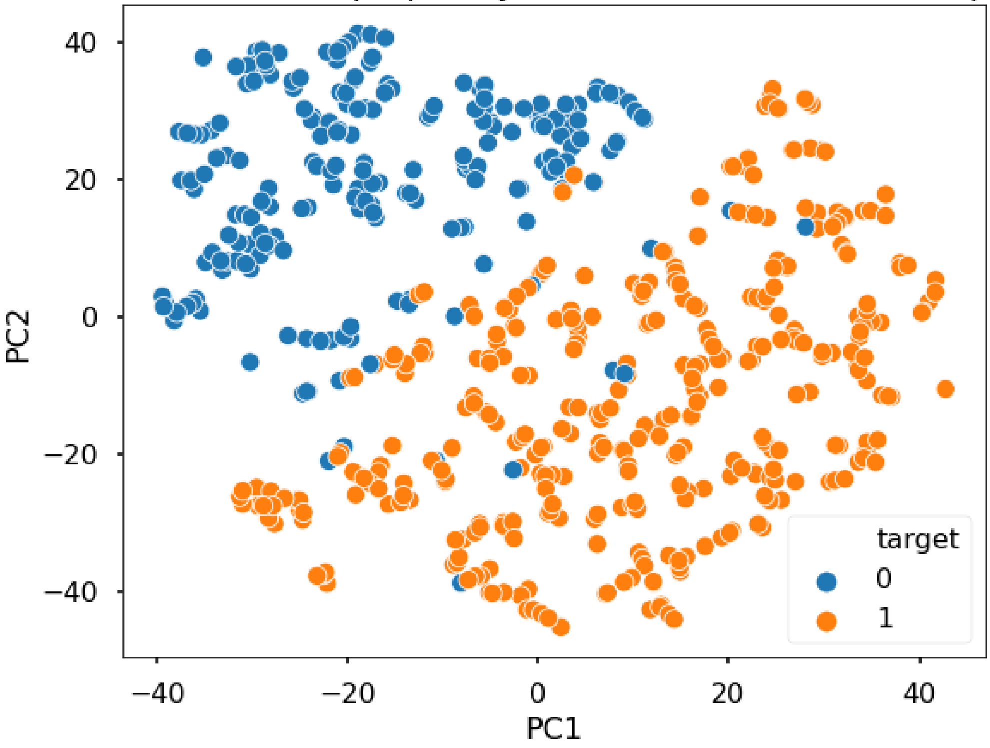

t-SNE visualization with perplexity -- 5, iterations -- 2000 and epsilon -- 10
target
0
1
PC1
PC2


t-SNE visualization with perplexity -- 5, iterations -- 3000 and epsilon -- 10
target
0
1
PC1
PC2

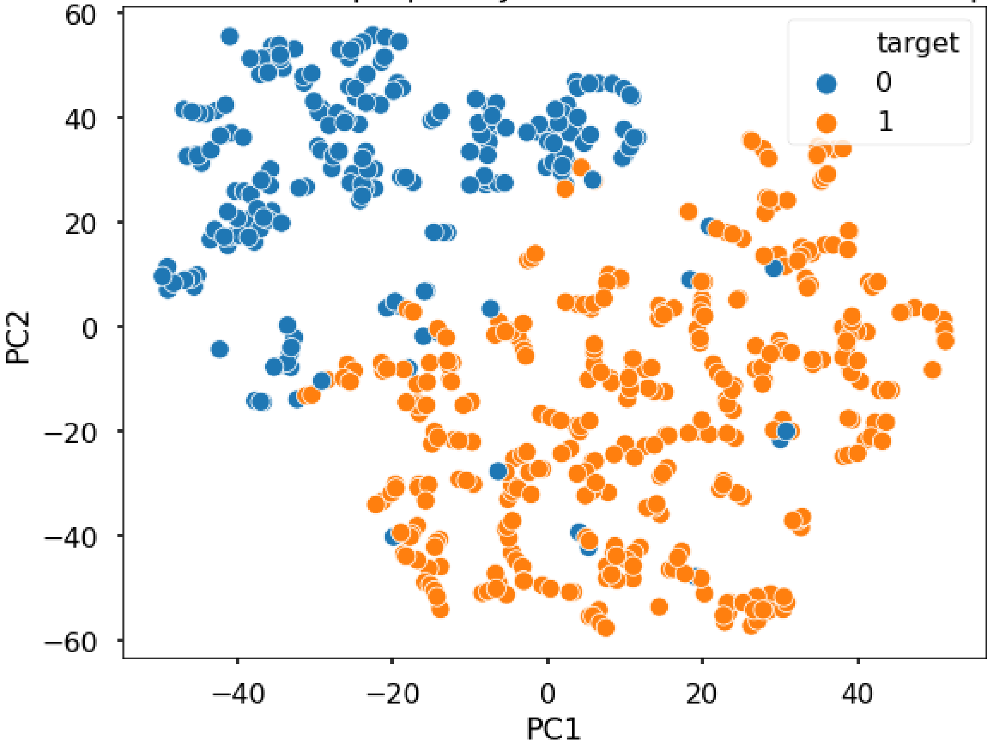

t-SNE visualization with perplexity -- 5, iterations -- 5000 and epsilon -- 10
target
0
1
PC2
PC1
60
40
20
0
−20
−40
−60
−80
−100
−75
−50
−25
25
50
75

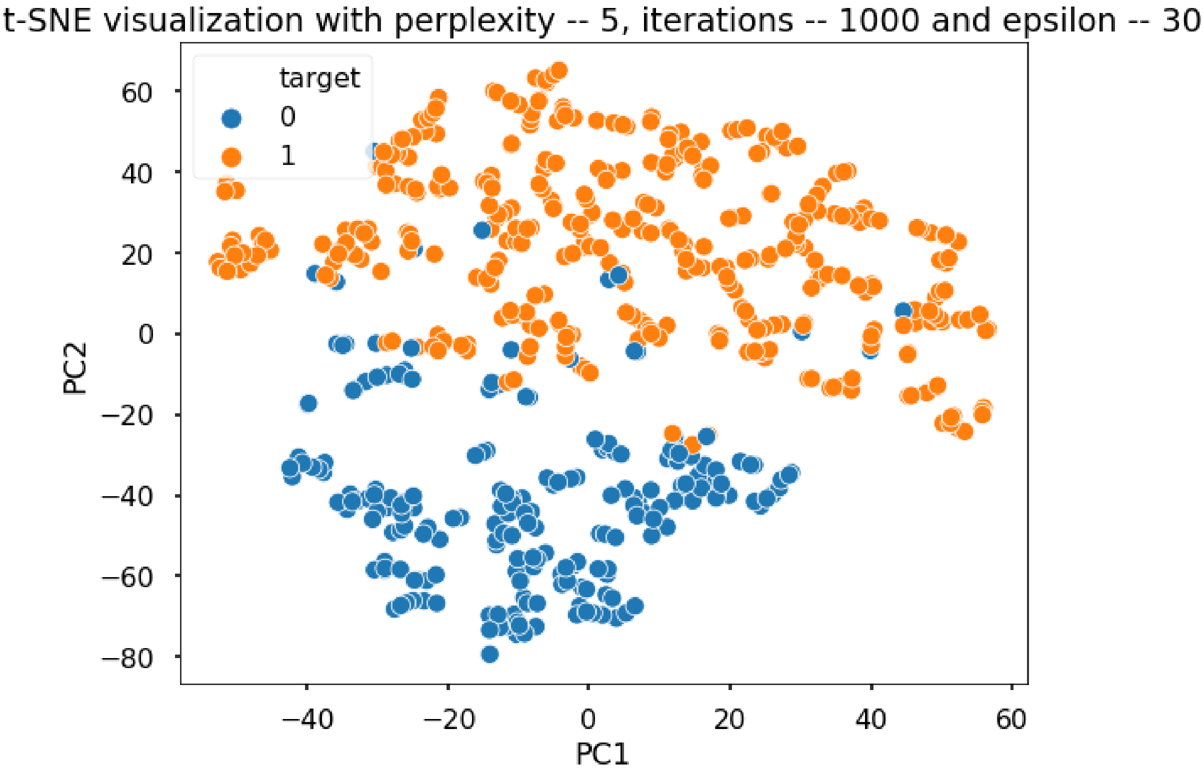

t-SNE visualization with perplexity -- 5, iterations -- 250 and epsilon -- 30
target
0
1
PC2
PC1
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−0.6
−0.4
−0.2
0.2
0.4
0.6

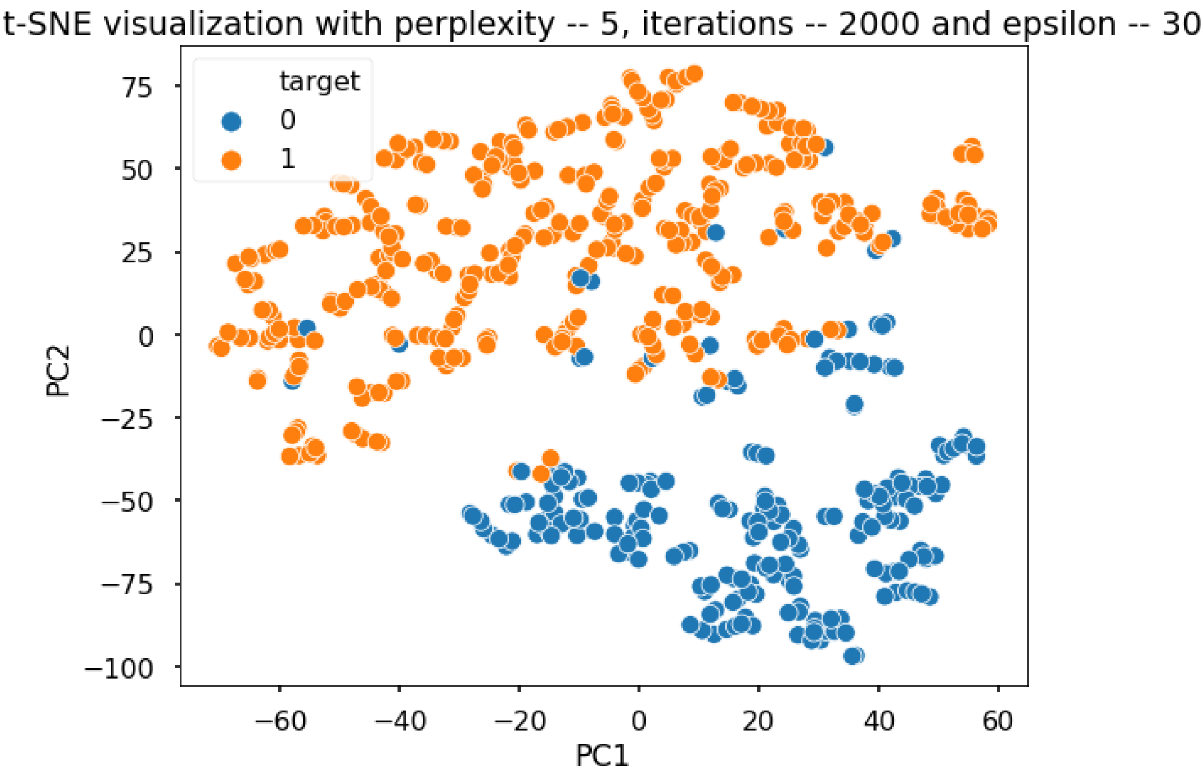
t-SNE visualization with perplexity -- 5, iterations -- 500 and epsilon -- 30



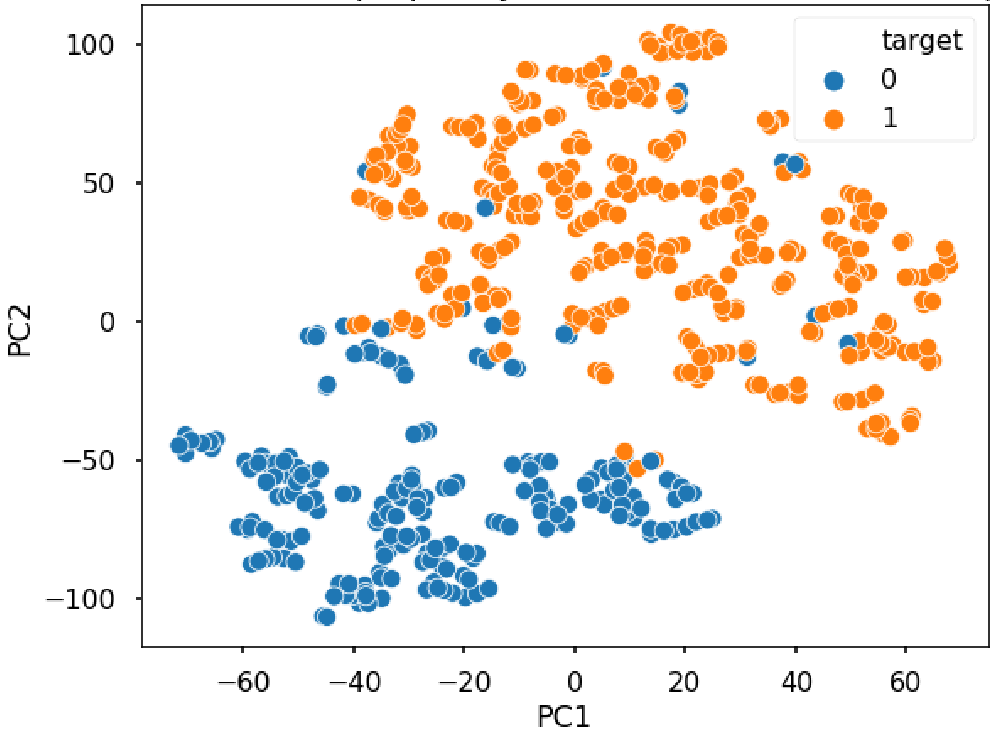
t-SNE visualization with perplexity -- 5, iterations -- 750 and epsilon -- 30



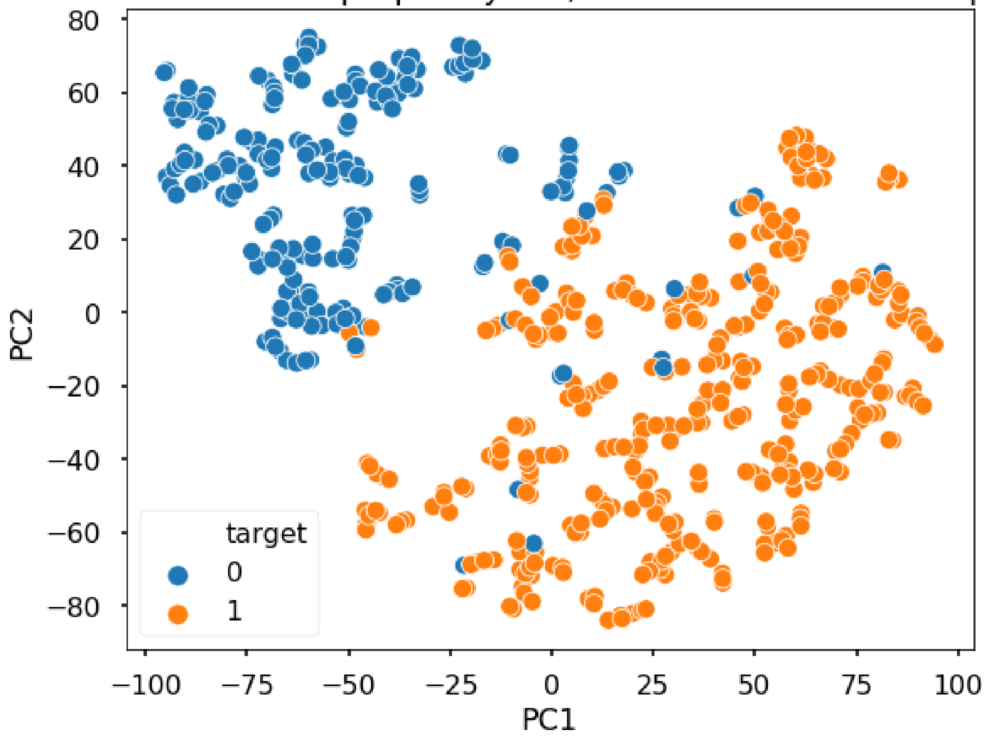

t-SNE visualization with perplexity -- 5, iterations -- 1000 and epsilon -- 30
target
0
1
PC2
PC1
60
40
20
0
−20
−40
−60
−80
−40
−20
0
20
40
60

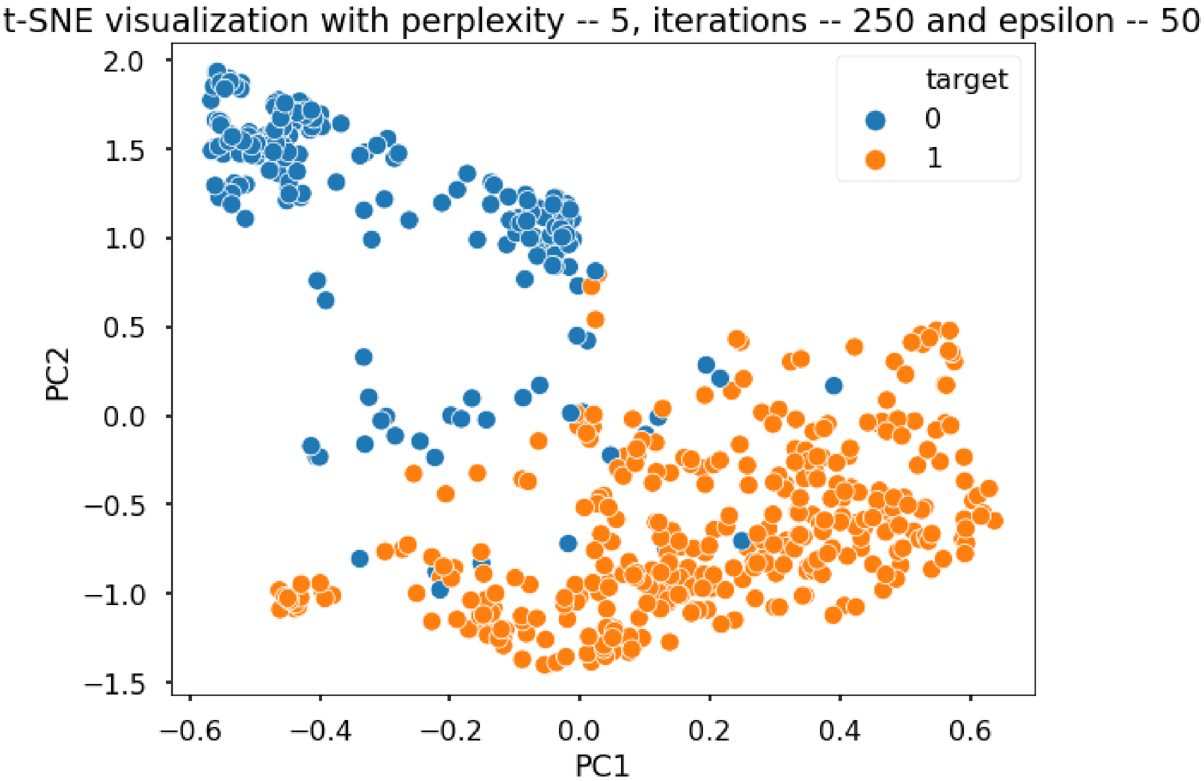

t-SNE visualization with perplexity -- 5, iterations -- 2000 and epsilon -- 30
target
0
1
PC2
PC1
75
50
25
0
−25
−50
−75
−100
−60
−40
−20
0
20
40
60

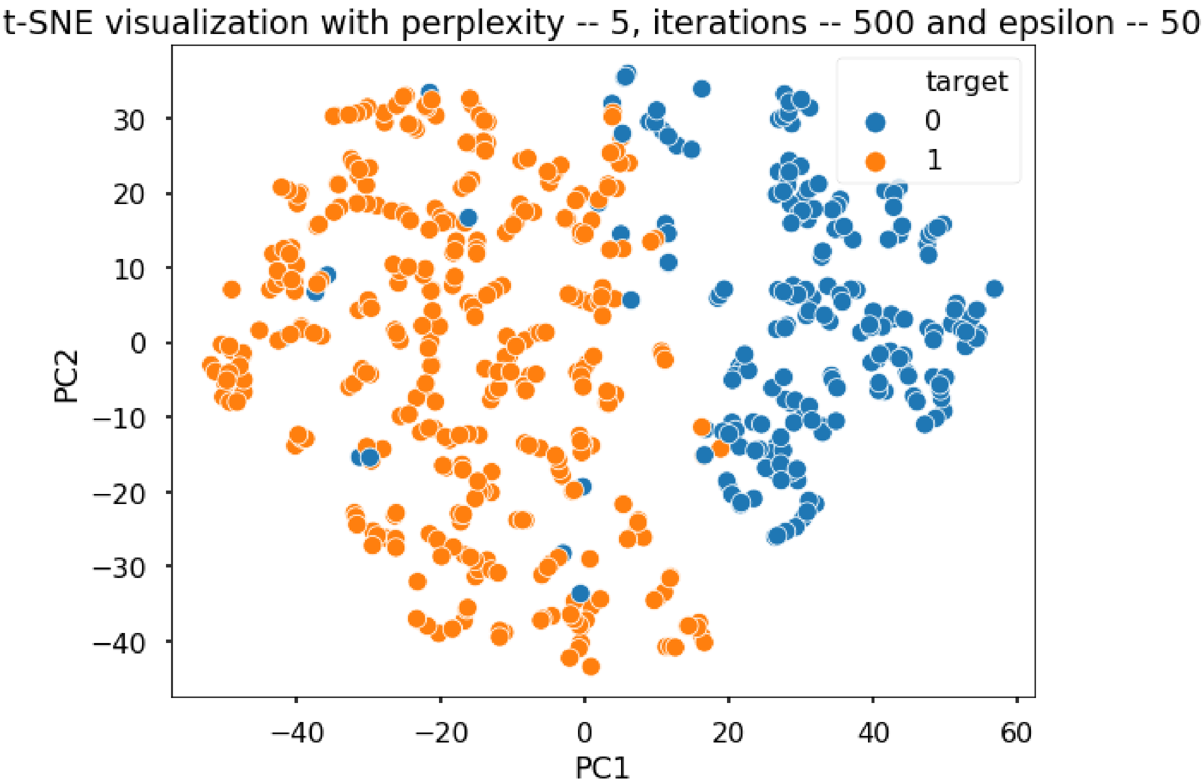
t-SNE visualization with perplexity -- 5, iterations -- 3000 and epsilon -- 30

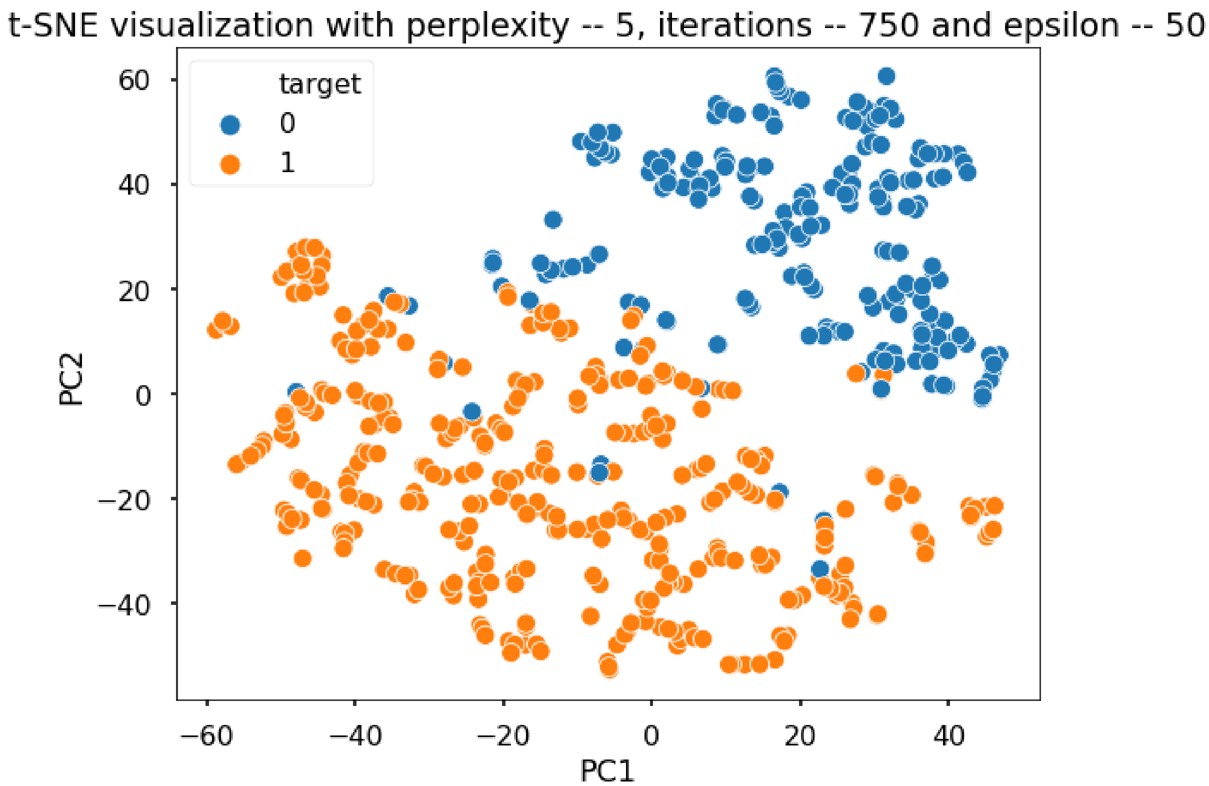


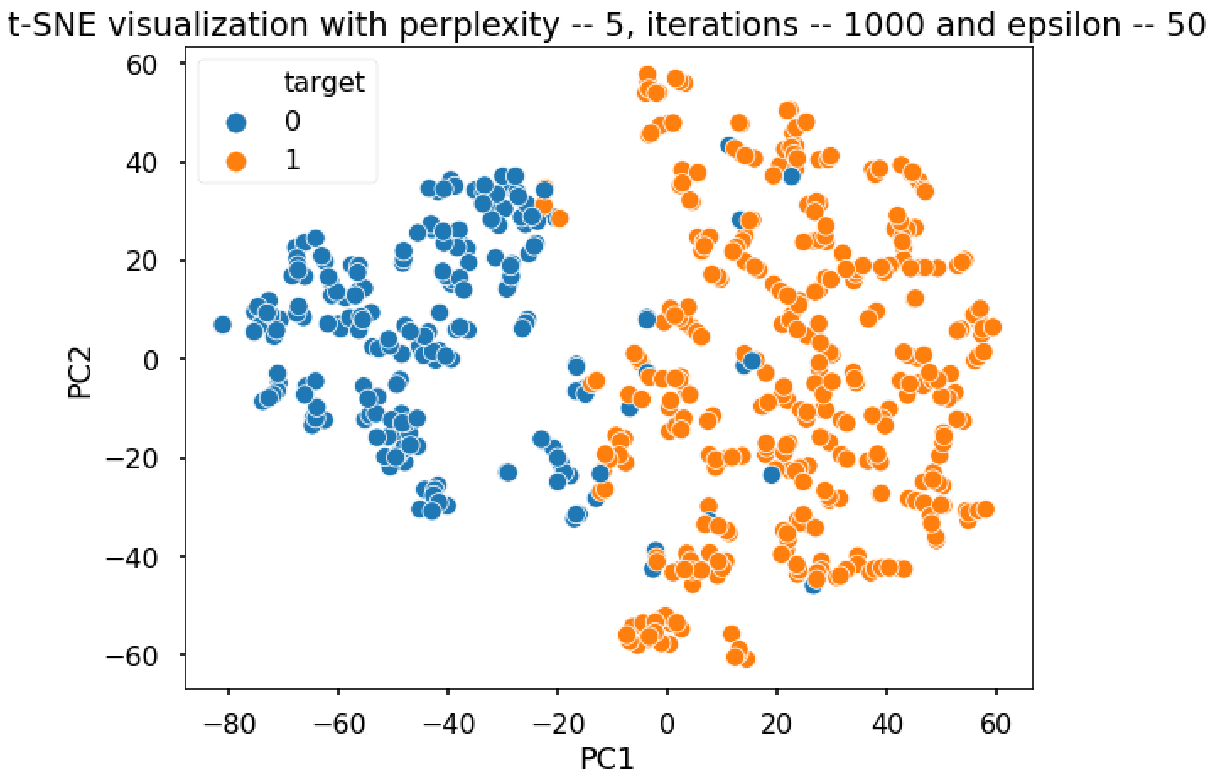
t-SNE visualization with perplexity -- 5, iterations -- 5000 and epsilon -- 30

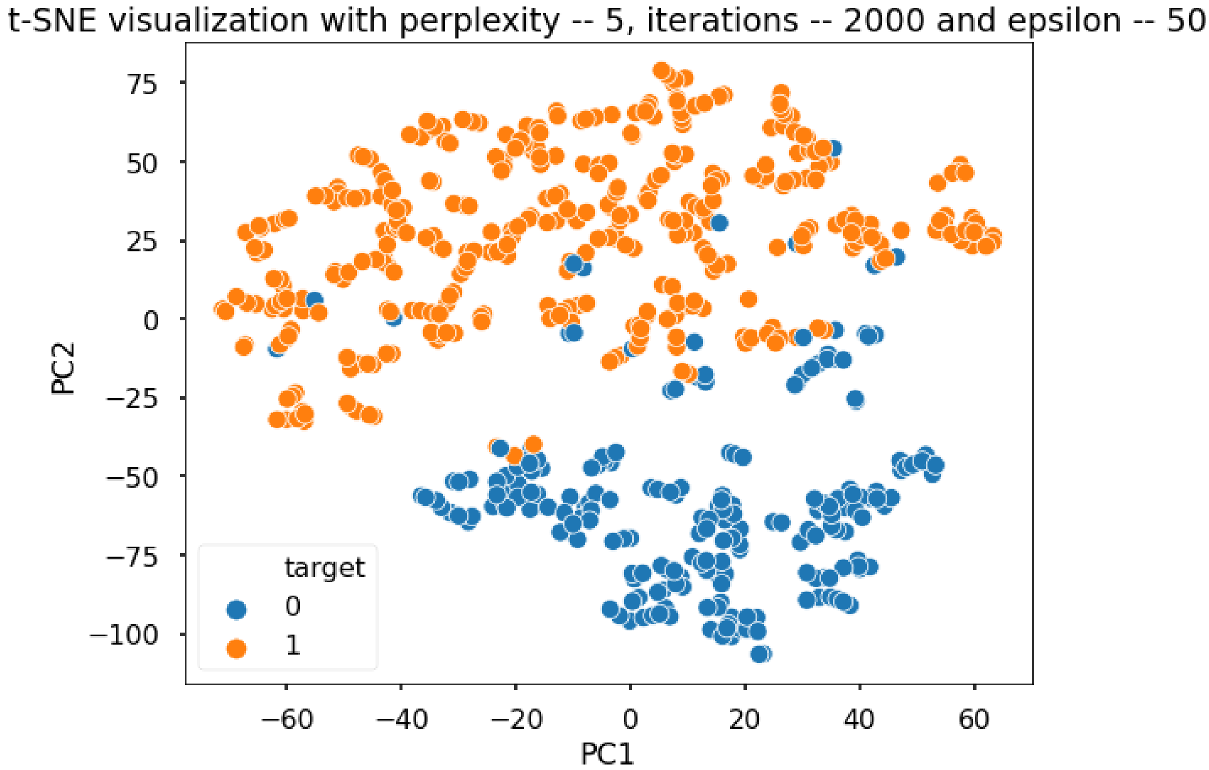


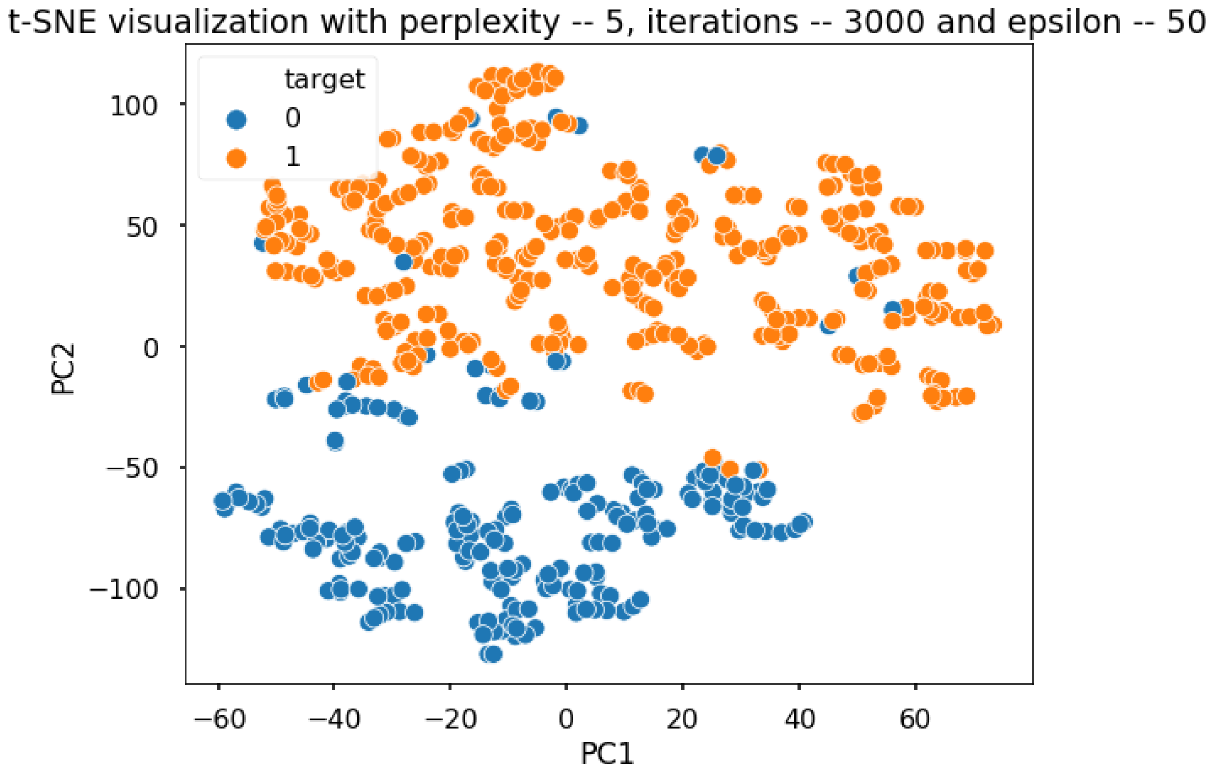

t-SNE visualization with perplexity -- 5, iterations -- 250 and epsilon -- 50
target
0
1
PC1
PC2
−0.6
−0.4
−0.2
0.0
0.2
0.4
0.6
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5


t-SNE visualization with perplexity -- 5, iterations -- 500 and epsilon -- 50
target
0
1
PC1
PC2
−40
−20
0
20
40
60
30
20
10
0
−10
−20
−30
−40

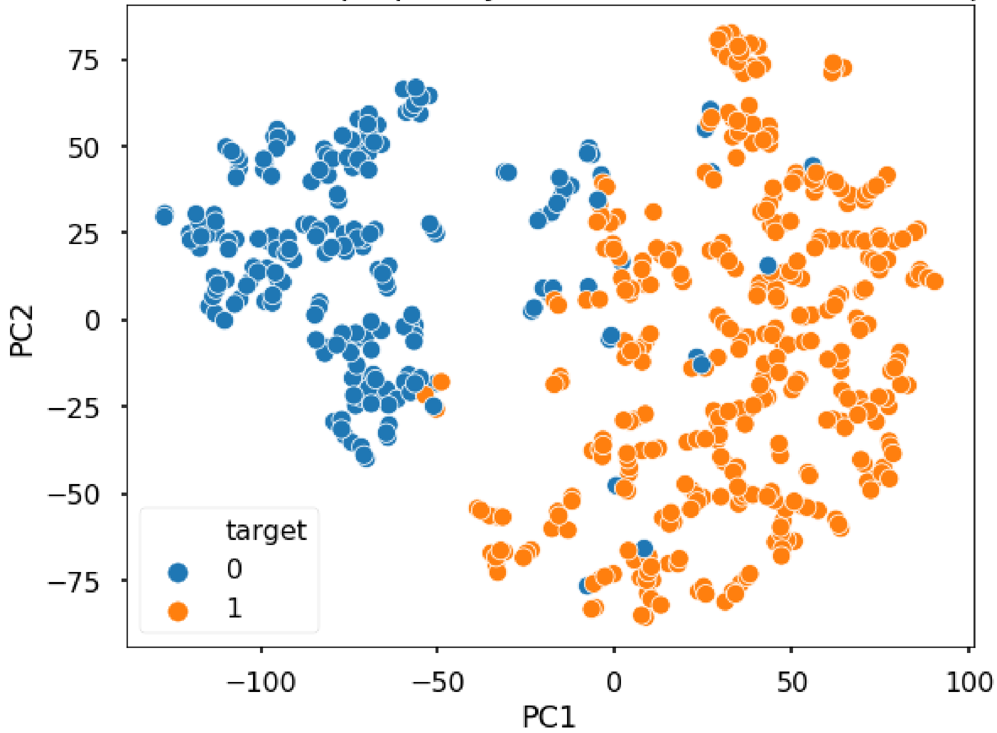

t-SNE visualization with perplexity -- 5, iterations -- 750 and epsilon -- 50
target
0
1
PC2
PC1
60
40
20
0
−20
−40
−60
−40
−20
0
20
40


t-SNE visualization with perplexity -- 5, iterations -- 1000 and epsilon -- 50
target
0
1
PC2
PC1
60
40
20
0
−20
−40
−60
−80
−60
−40
−20
0
20
40
60

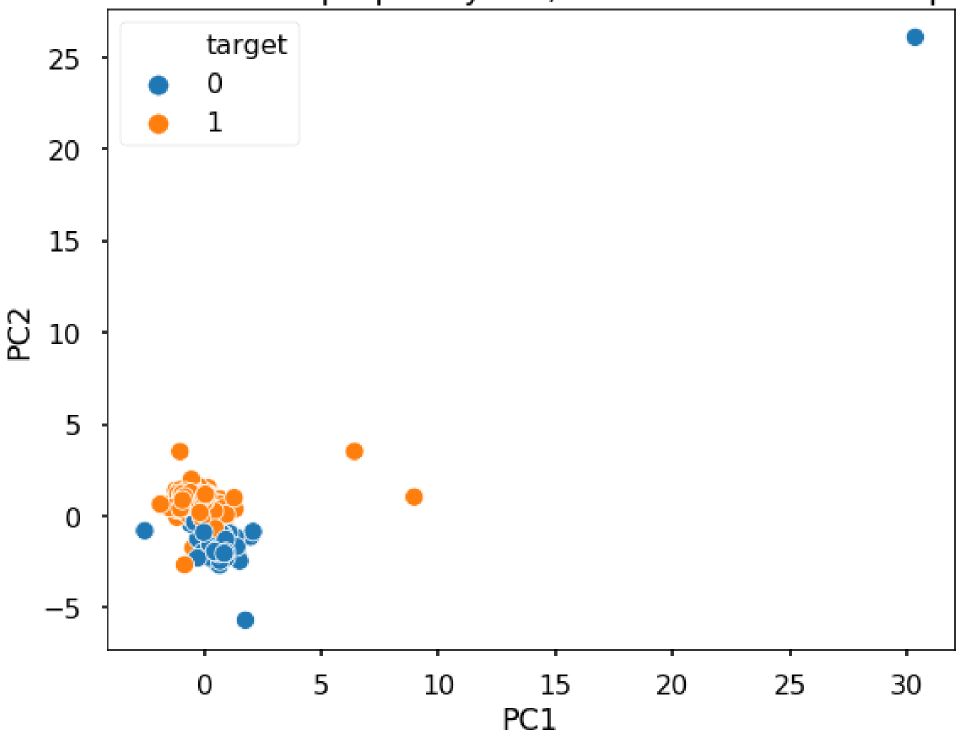

t-SNE visualization with perplexity -- 5, iterations -- 2000 and epsilon -- 50
PC2
PC1
target
0
1
75
50
25
0
−25
−50
−75
−100
−60
−40
−20
0
20
40
60

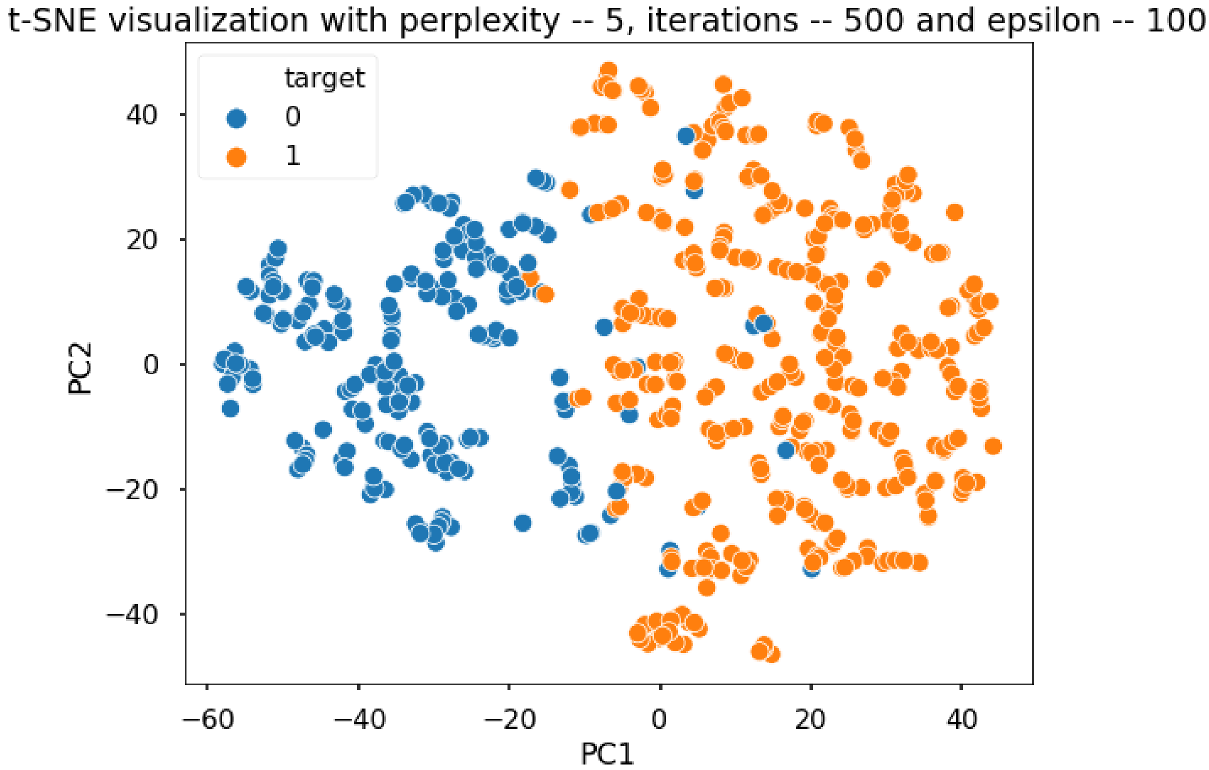

t-SNE visualization with perplexity -- 5, iterations -- 3000 and epsilon -- 50
PC2
PC1
target
0
1
100
50
0
−50
−100
−60
−40
−20
0
20
40
60

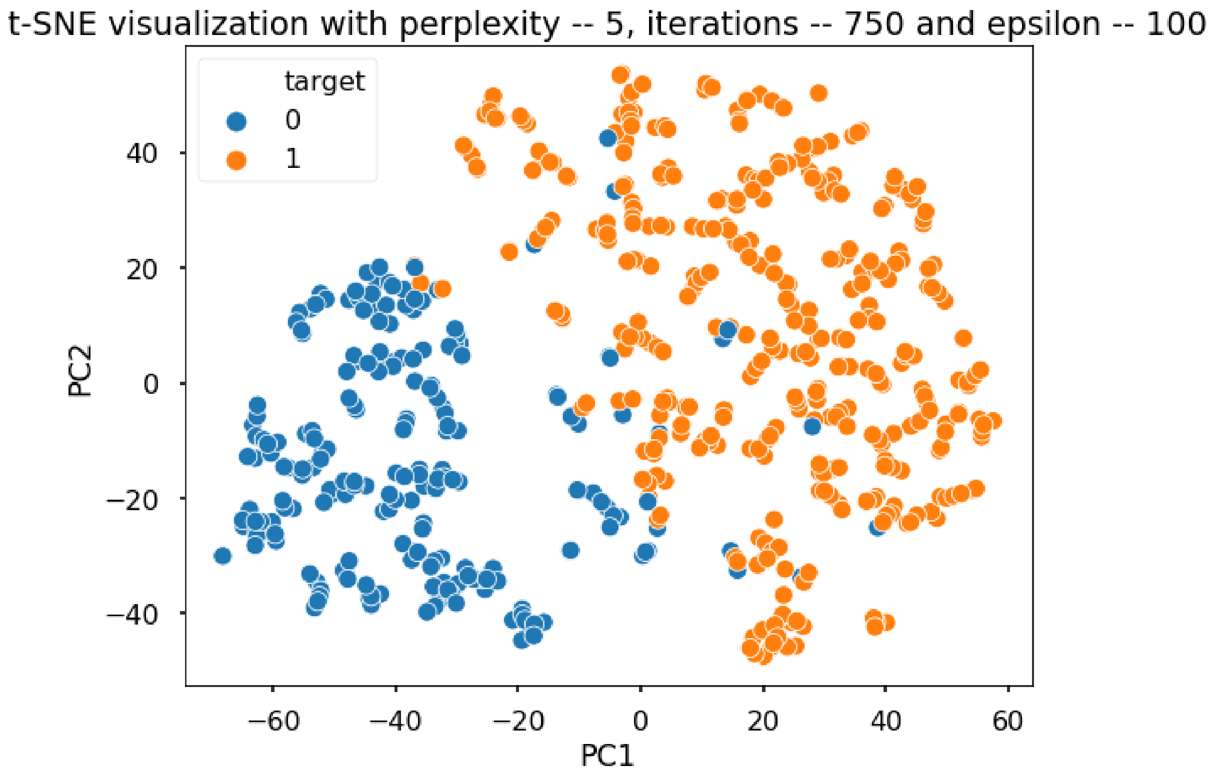
t-SNE visualization with perplexity -- 5, iterations -- 5000 and epsilon -- 50

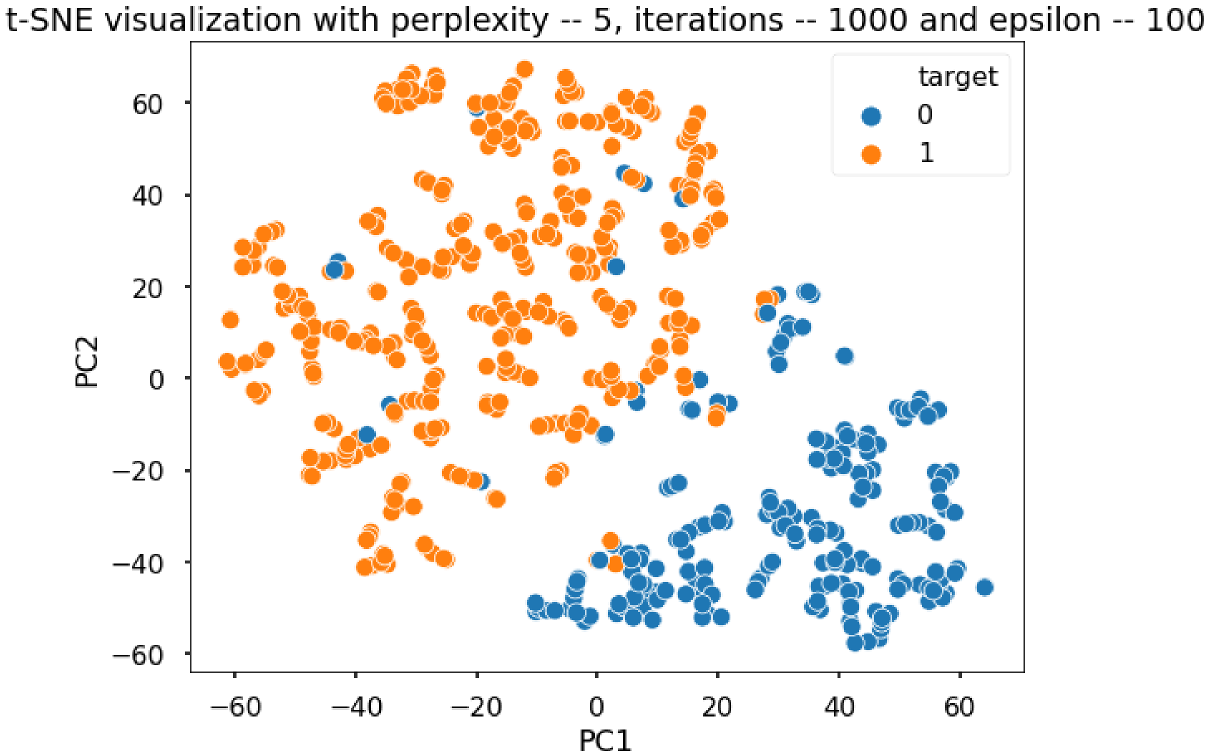


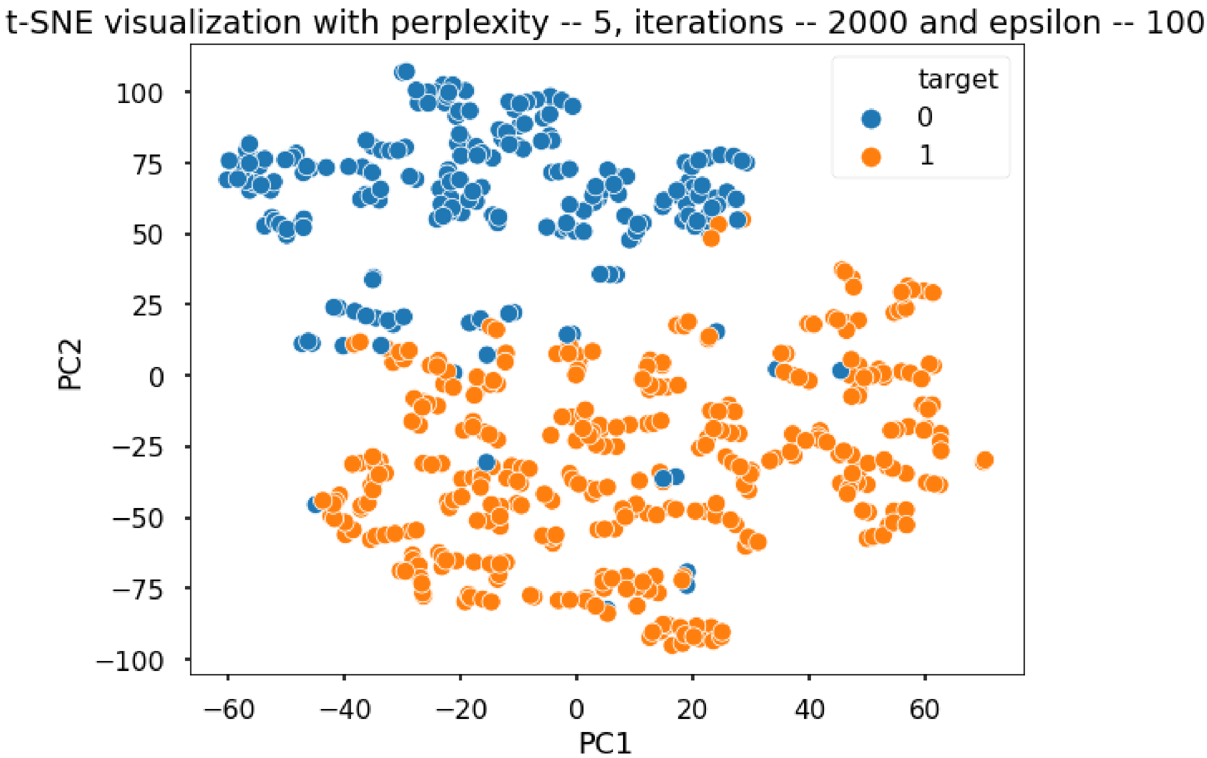
t-SNE visualization with perplexity -- 5, iterations -- 250 and epsilon -- 100

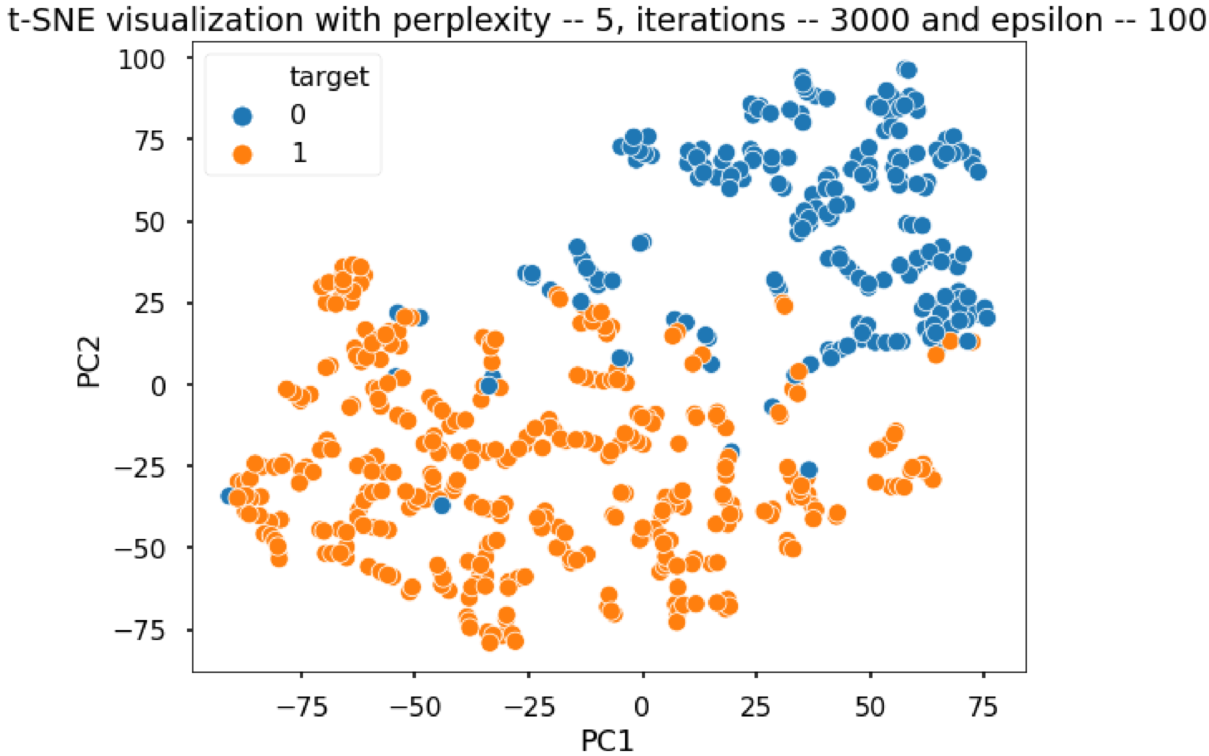


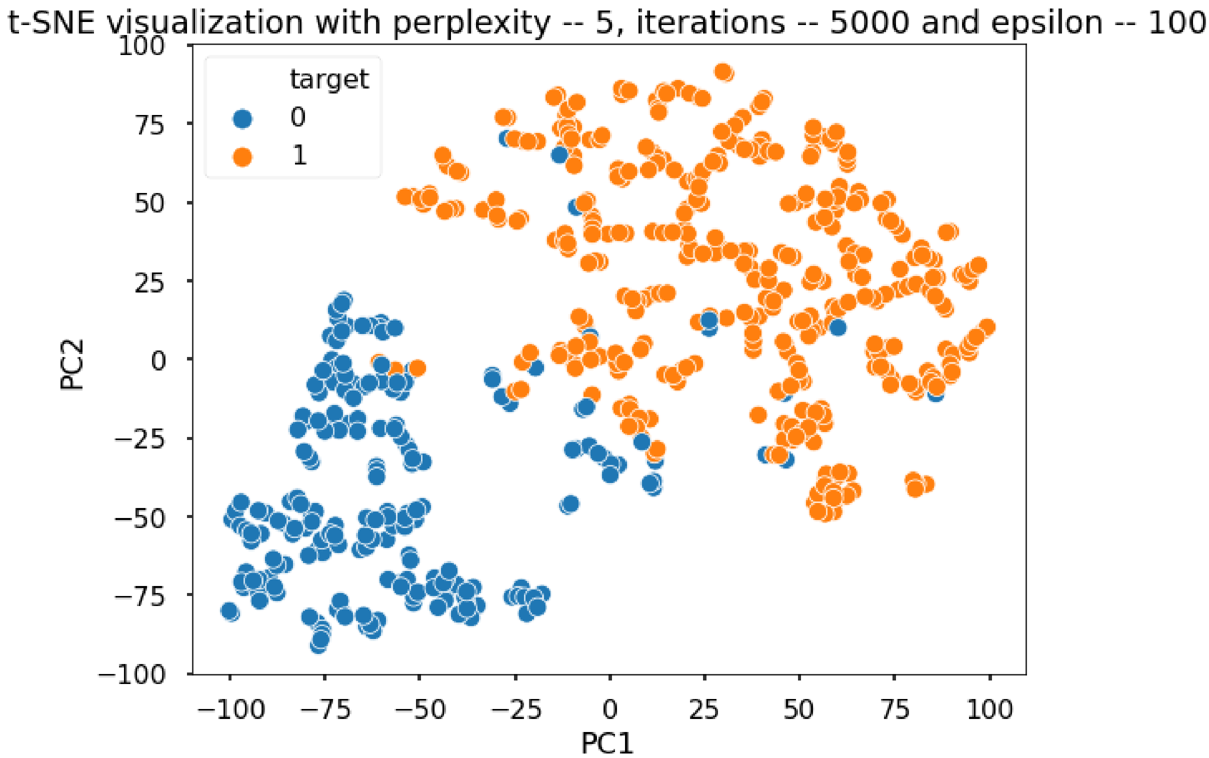

t-SNE visualization with perplexity -- 5, iterations -- 500 and epsilon -- 100
target
0
1
PC2
PC1
40
20
0
−20
−40
−60
−40
−20
0
20
40

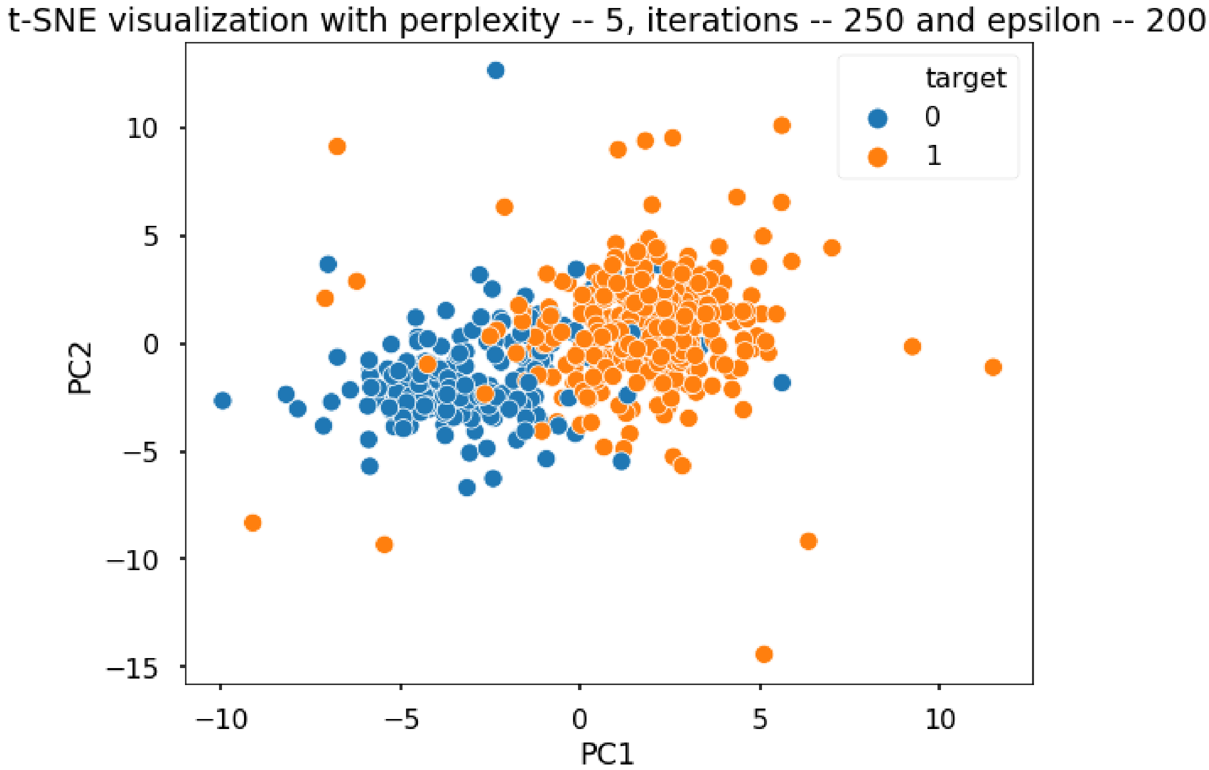

t-SNE visualization with perplexity -- 5, iterations -- 750 and epsilon -- 100
target
0
1
PC2
PC1
40
20
0
−20
−40
−60
−40
−20
0
20
40
60


t-SNE visualization with perplexity -- 5, iterations -- 1000 and epsilon -- 100
target
0
1
PC2
PC1
60
40
20
0
−20
−40
−60
−60
−40
−20
0
20
40
60

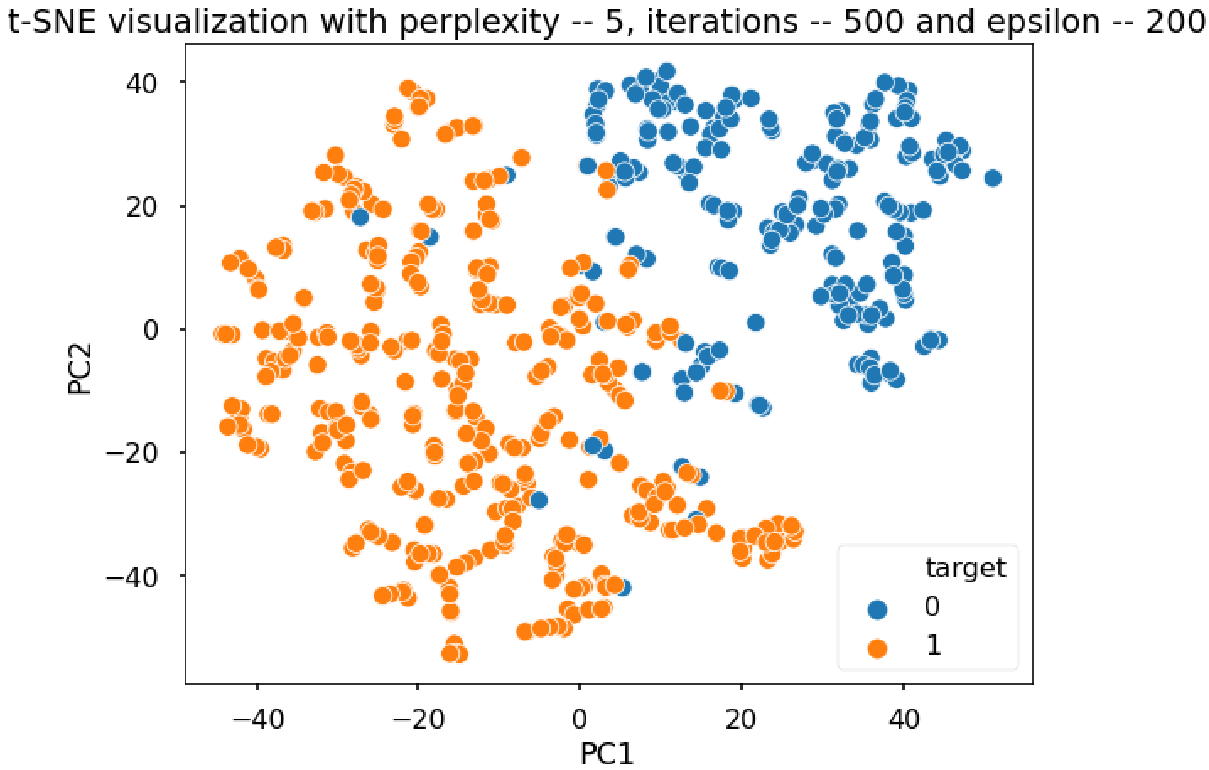

t-SNE visualization with perplexity -- 5, iterations -- 2000 and epsilon -- 100
target
0
1
PC2
PC1
100
75
50
25
0
−25
−50
−75
−100
−60
−40
−20
0
20
40
60


t-SNE visualization with perplexity -- 5, iterations -- 3000 and epsilon -- 100
target
0
1
PC2
PC1
100
75
50
25
0
−25
−50
−75
−75
−50
−25
0
25
50
75


t-SNE visualization with perplexity -- 5, iterations -- 5000 and epsilon -- 100
target
0
1
PC2
PC1
100
75
50
25
0
−25
−50
−75
−100
−100
−75
−50
−25
0
25
50
75
100

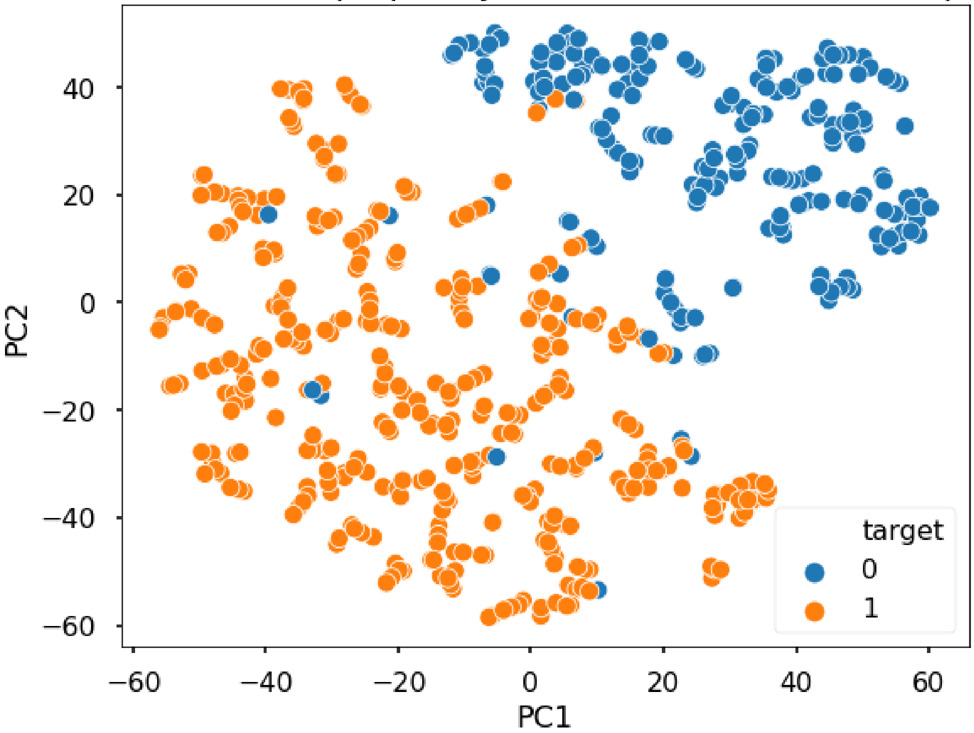


t-SNE visualization with perplexity -- 5, iterations -- 250 and epsilon -- 200
target
0
1
PC2
PC1
10
5
0
−5
−10
−15
−10
−5
0
5
10

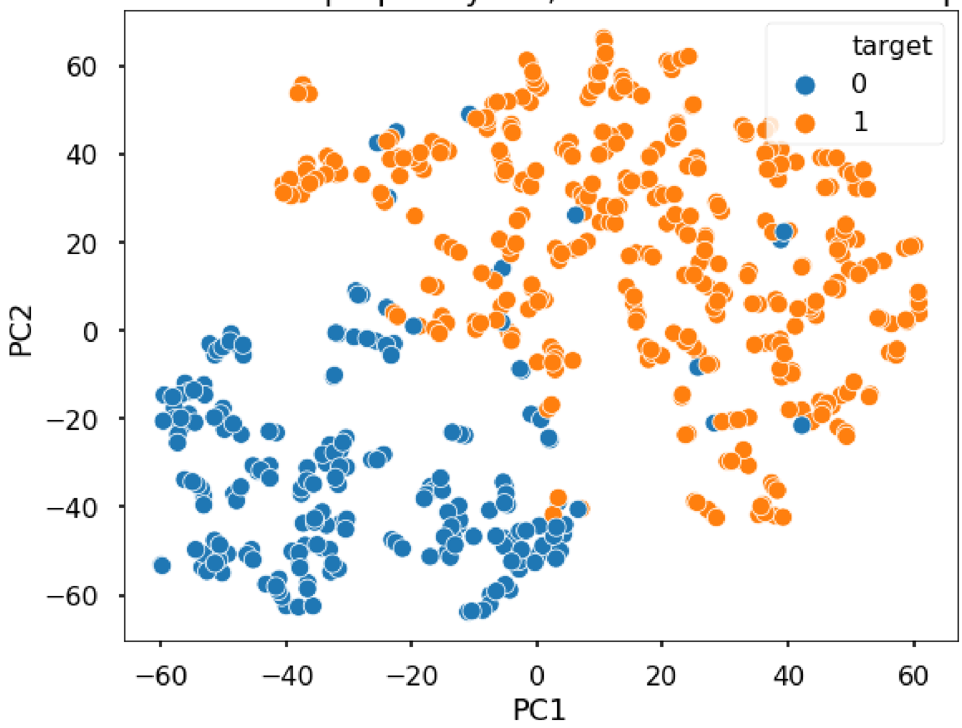


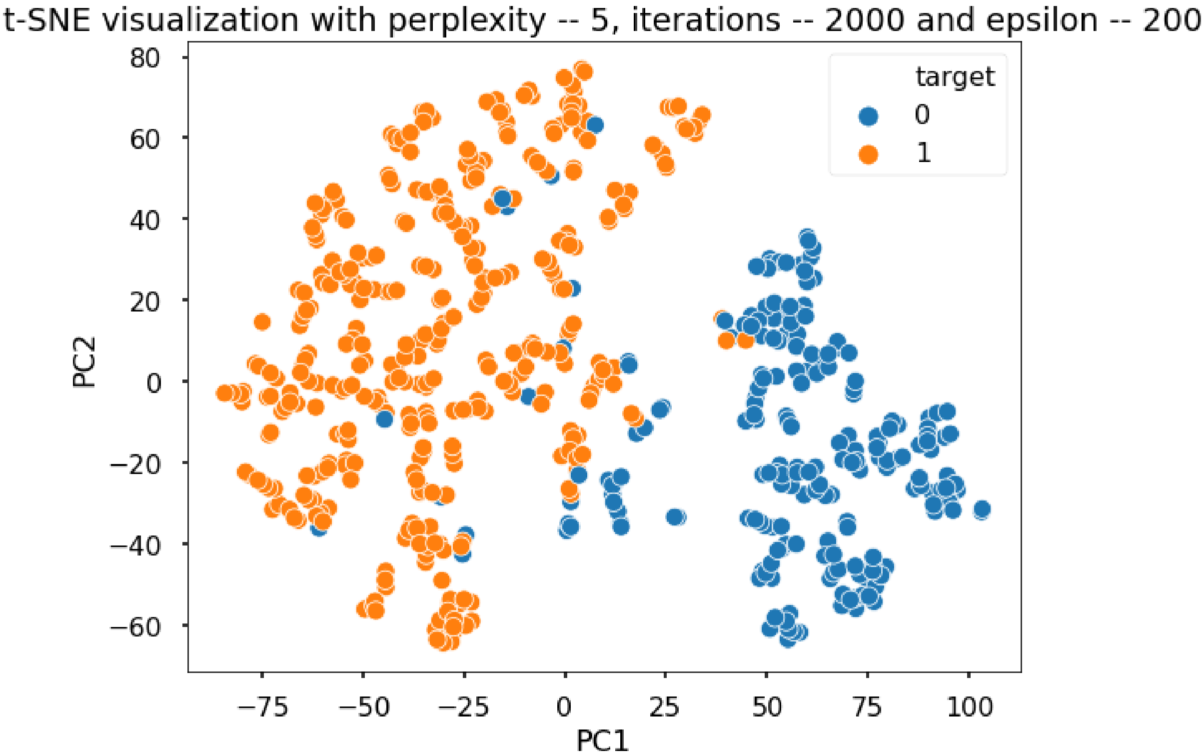
t-SNE visualization with perplexity -- 5, iterations -- 500 and epsilon -- 200
target
0
1
PC2
PC1
40
20
0
−20
−40
−40
−20
0
20
40

t-SNE visualization with perplexity -- 5, iterations -- 750 and epsilon -- 200

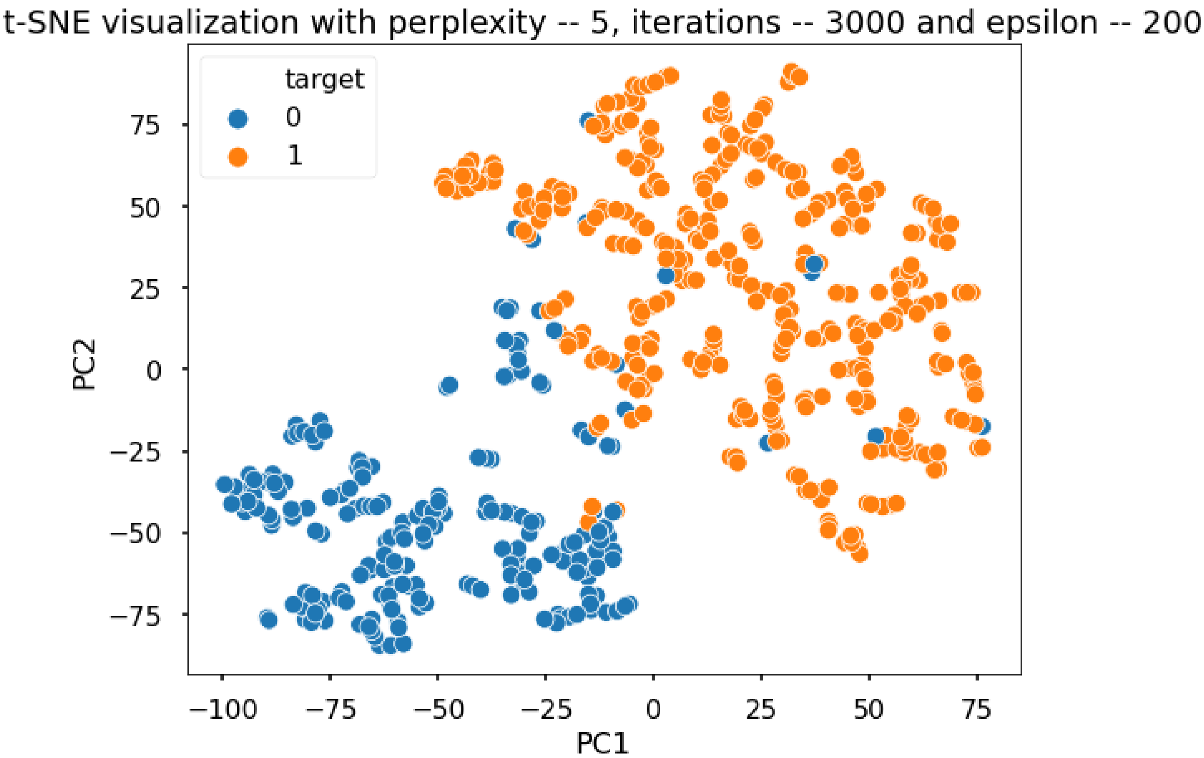


t-SNE visualization with perplexity -- 5, iterations -- 1000 and epsilon -- 200

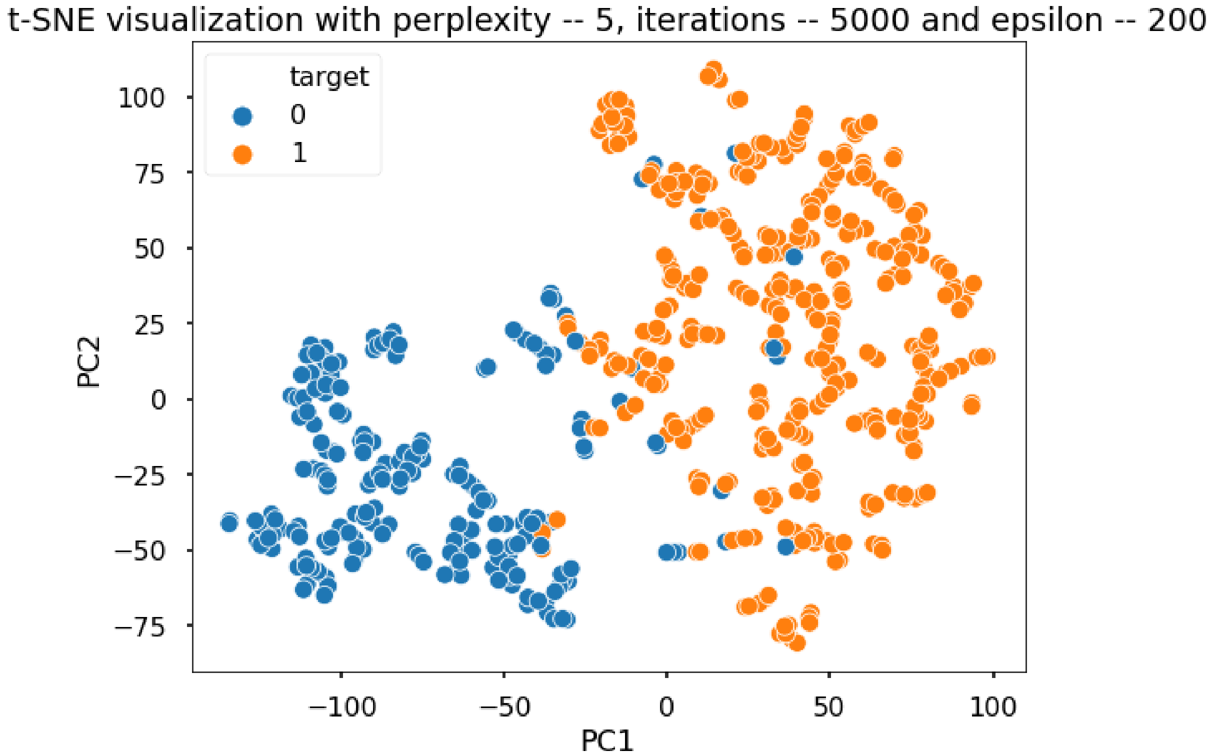


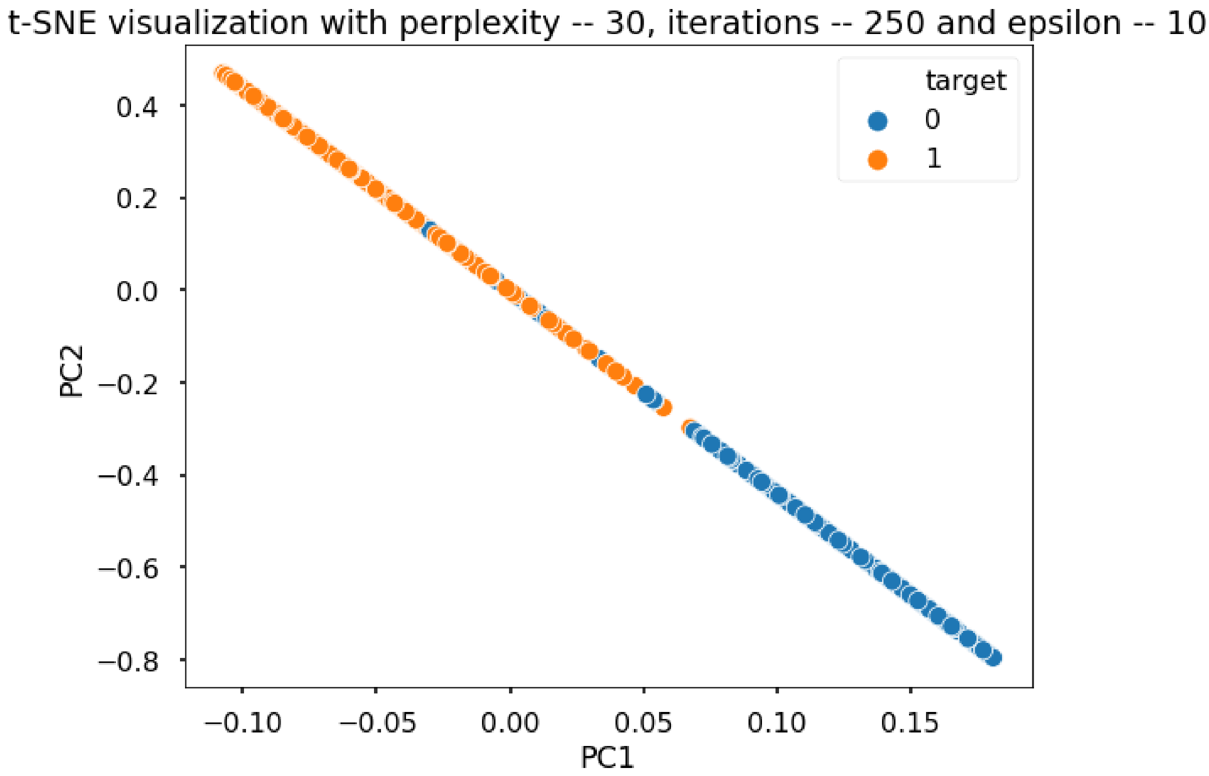


t-SNE visualization with perplexity -- 5, iterations -- 2000 and epsilon -- 200
target
0
1
PC2
PC1
80
60
40
20
0
−20
−40
−60
−75
−50
−25
0
25
50
75
100

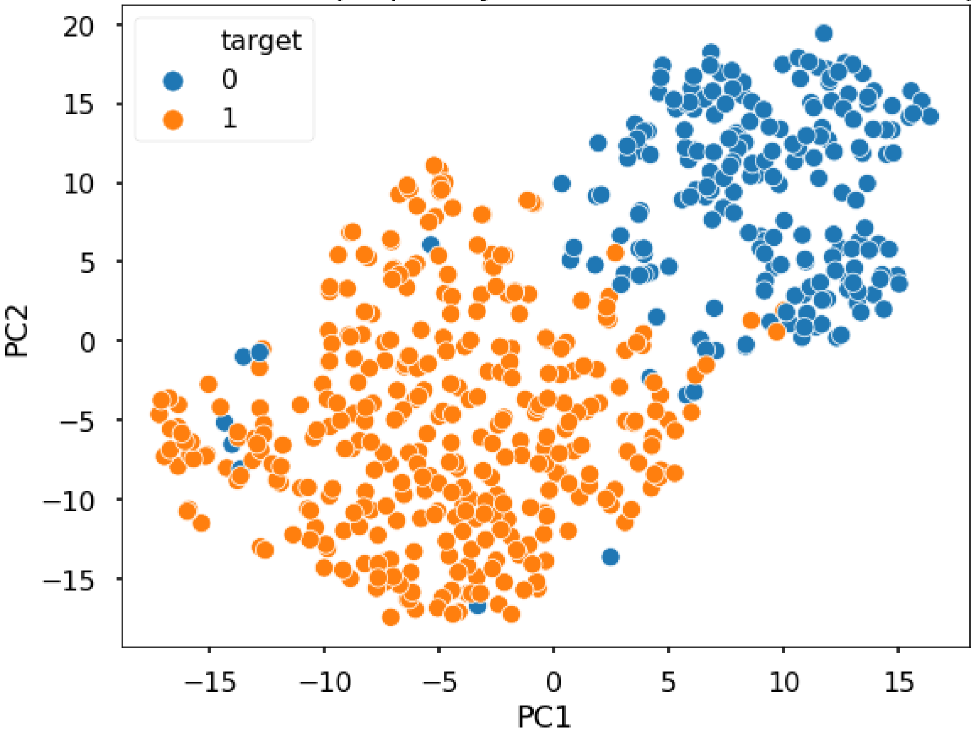


t-SNE visualization with perplexity -- 5, iterations -- 3000 and epsilon -- 200
target
0
1
PC2
PC1
75
50
25
0
−25
−50
−75
−100
−75
−50
−25
0
25
50
75

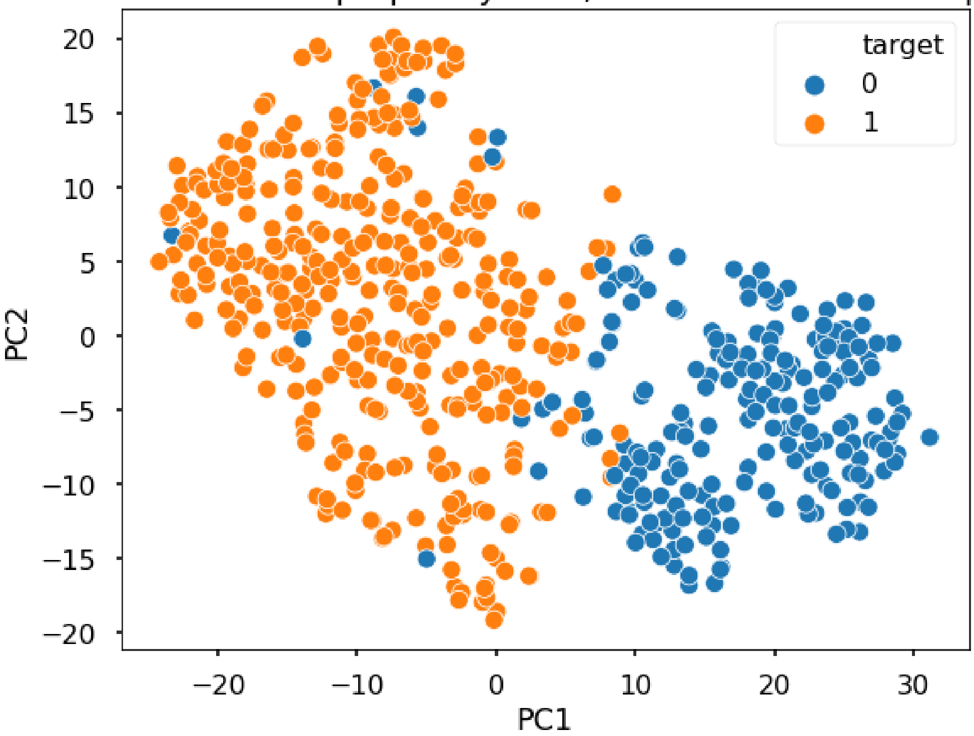

t-SNE visualization with perplexity -- 5, iterations -- 5000 and epsilon -- 200
target
0
1
100
75
50
25
0
−25
−50
−75
PC2
−100
−50
0
50
100
PC1

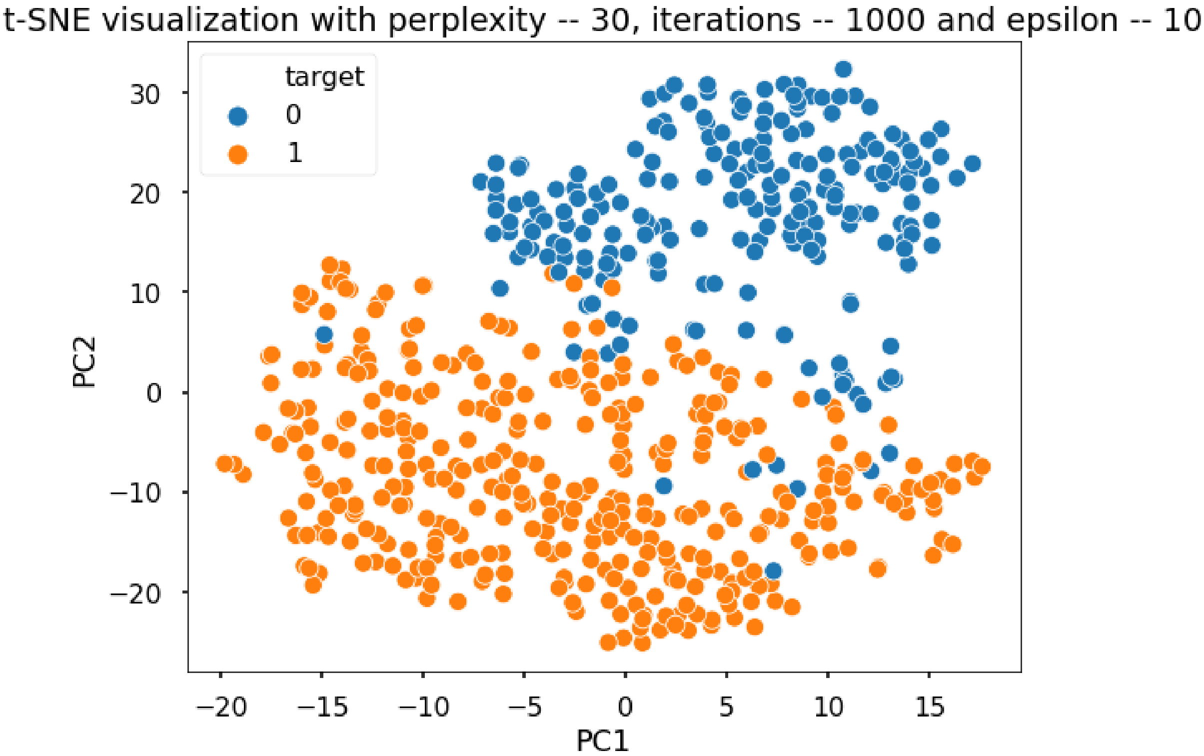

t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 10
target
0
1
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
PC2
−0.10
−0.05
0.00
0.05
0.10
0.15
PC1

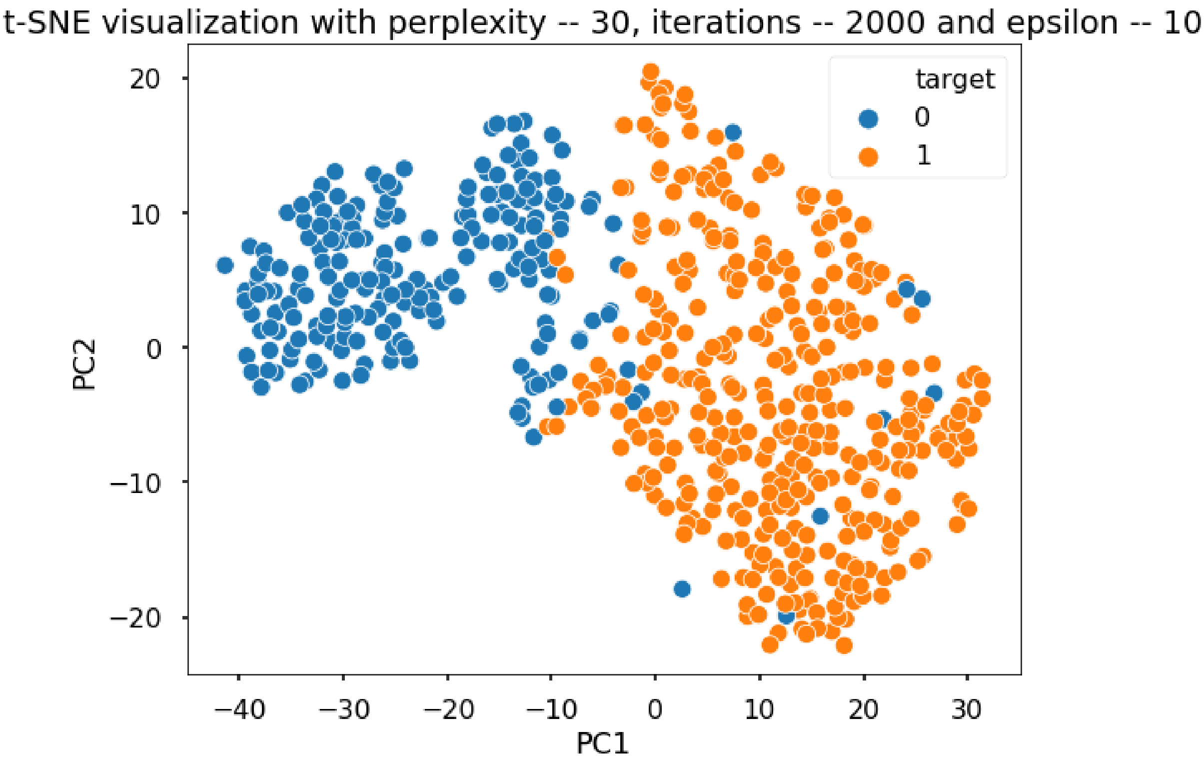
t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 10



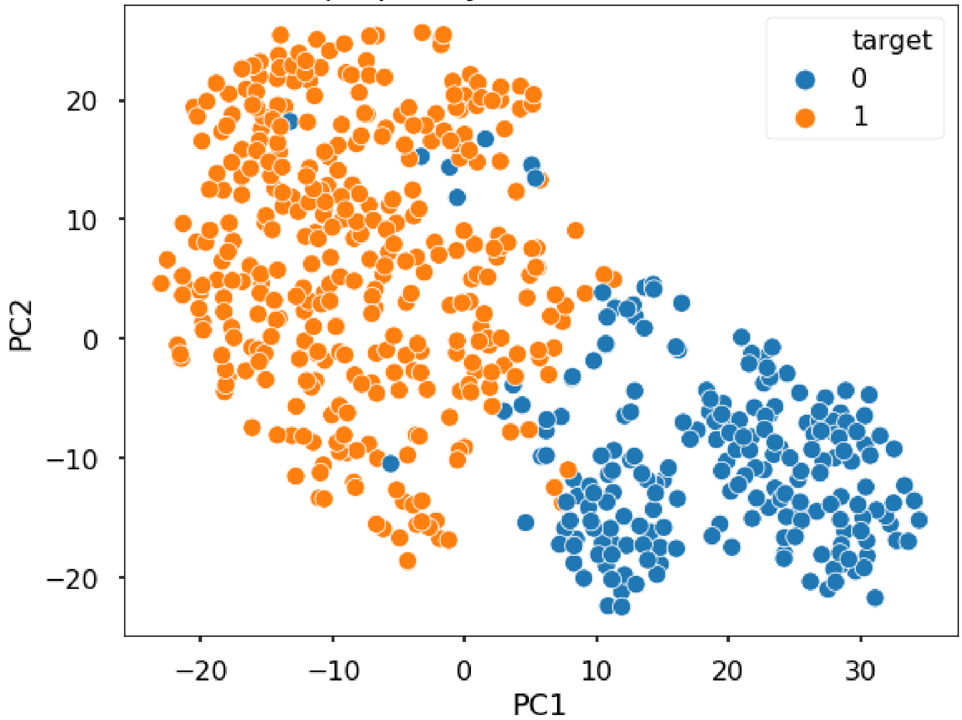
t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 10



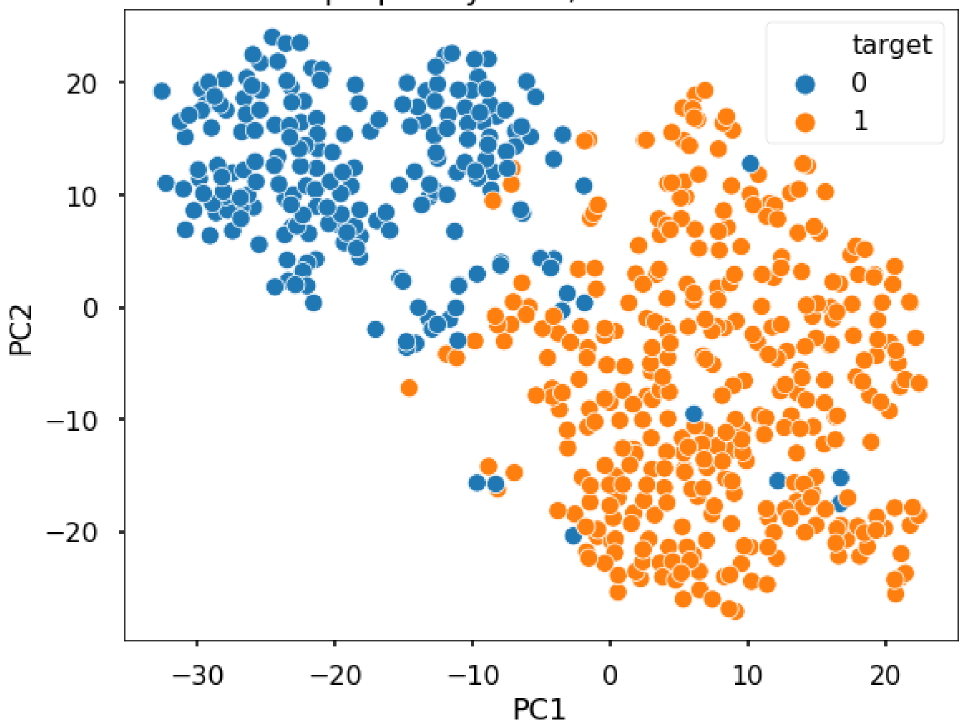

t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 10
target
0
1
PC2
PC1
30
20
10
0
−10
−20
−20
−15
−10
−5
0
5
10
15

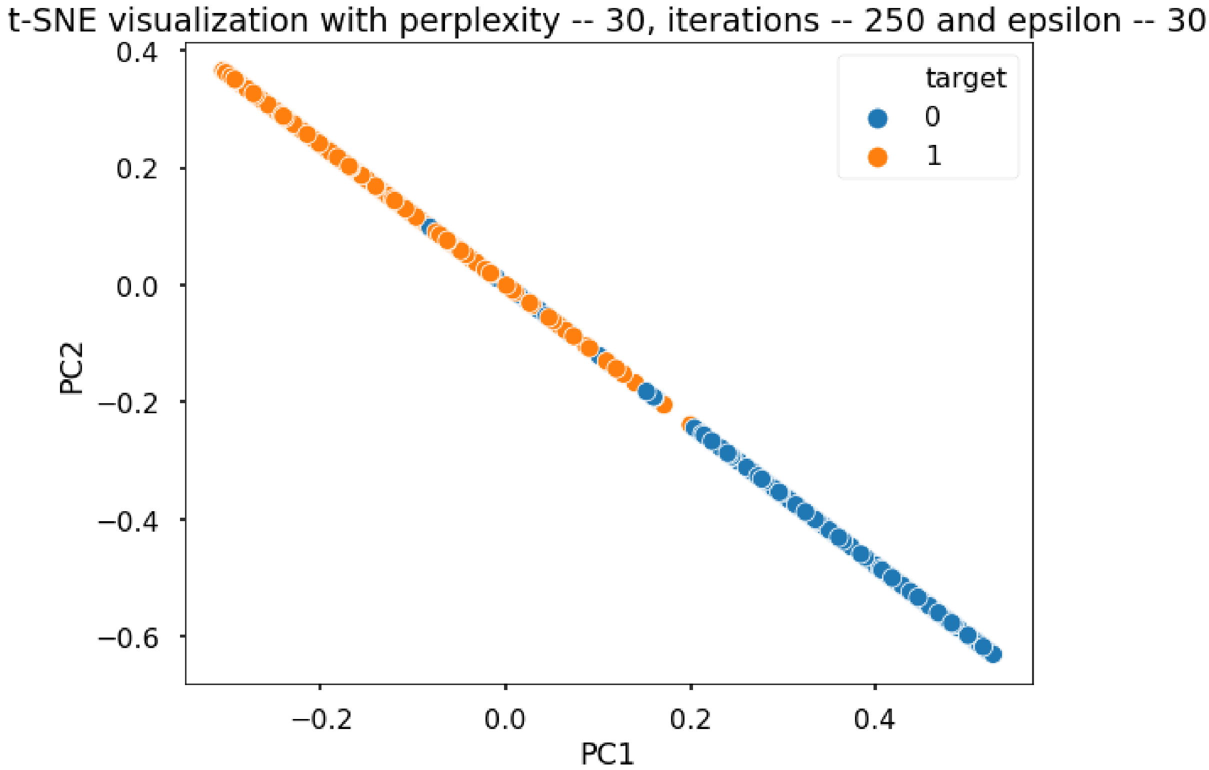

t-SNE visualization with perplexity -- 30, iterations -- 2000 and epsilon -- 10
target
0
1
PC2
PC1
20
10
0
−10
−20
−40
−30
−20
−10
0
10
20
30

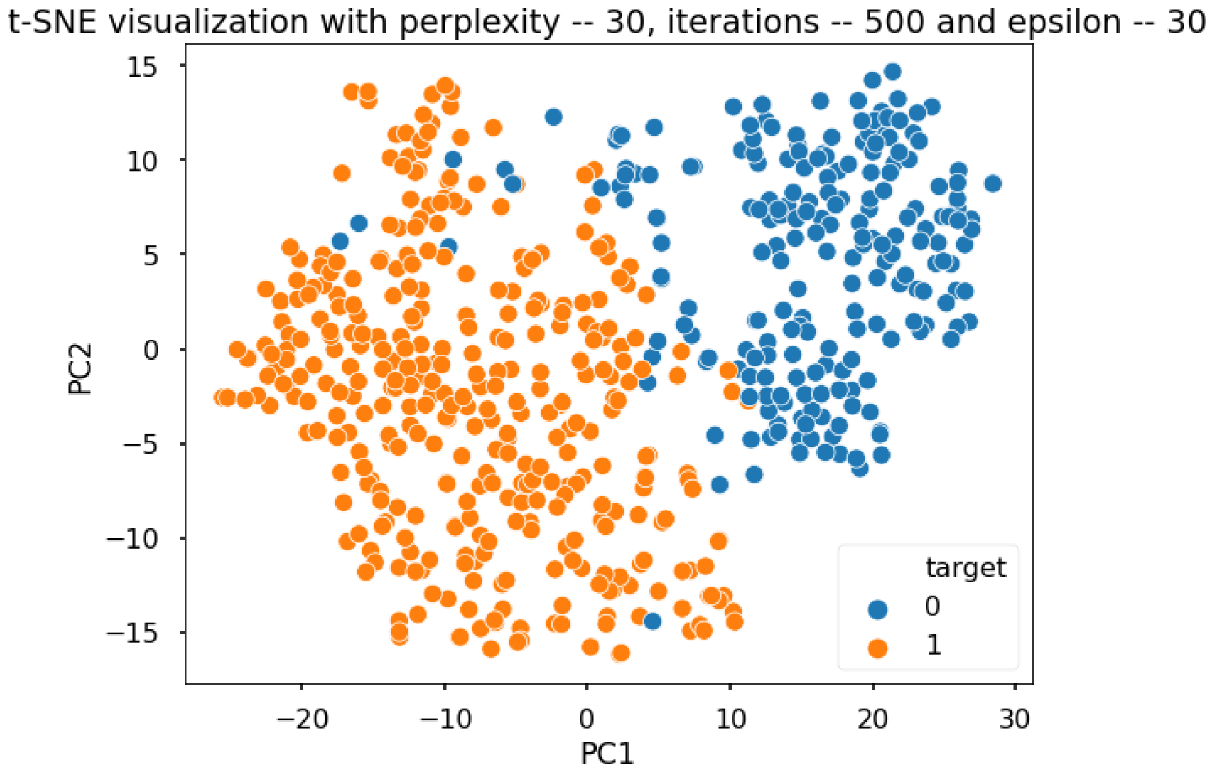
t-SNE visualization with perplexity -- 30, iterations -- 3000 and epsilon -- 10



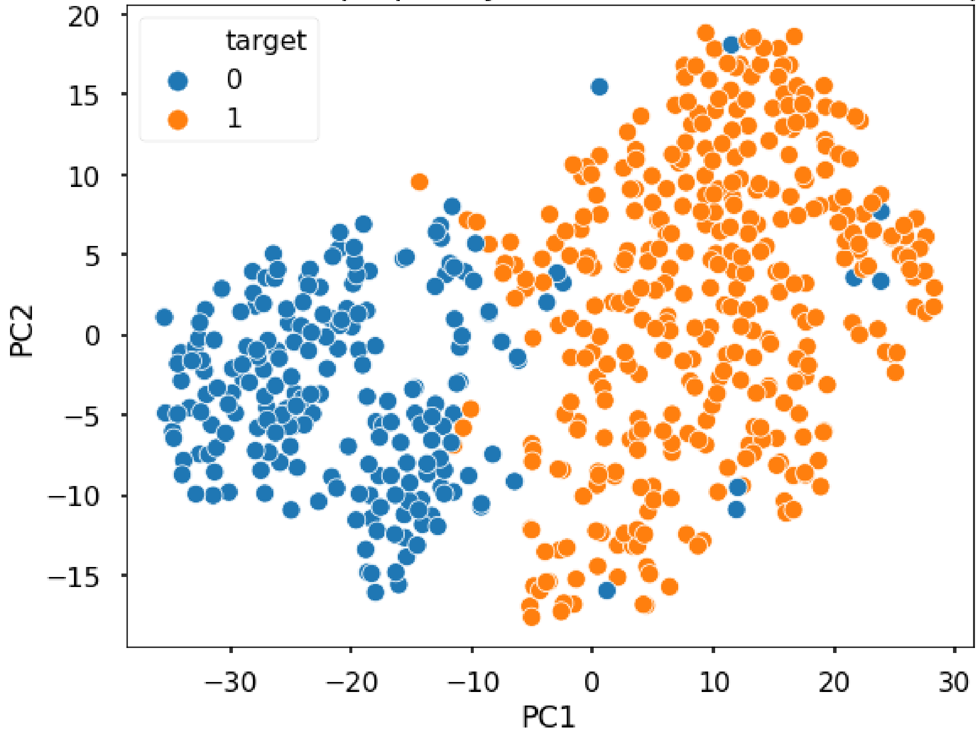
t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 10



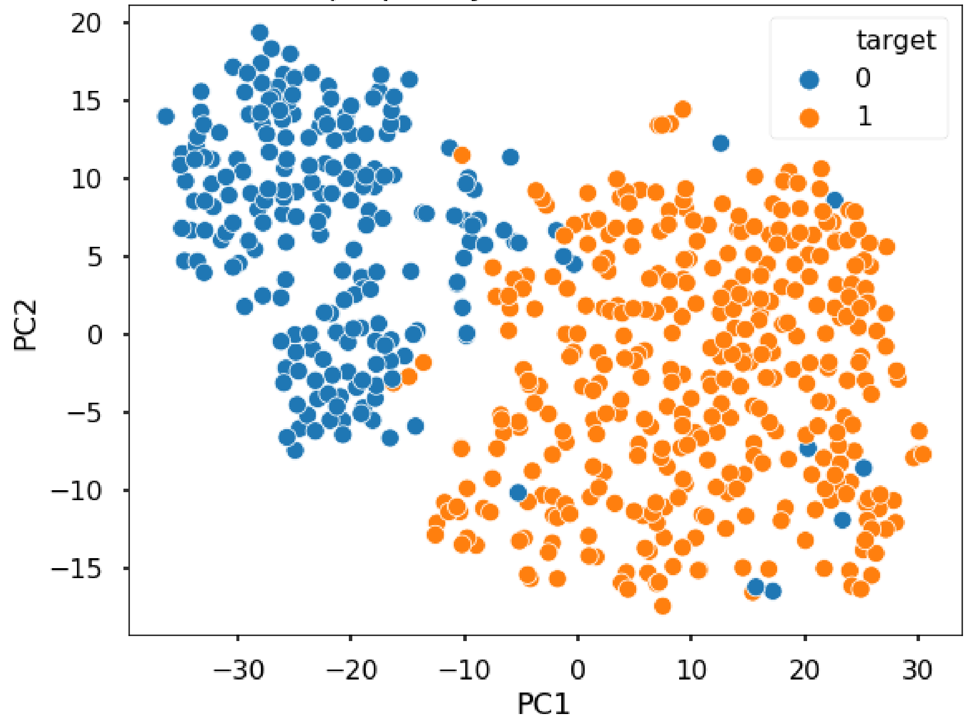

t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 30
target
0
1
PC1
PC2
−0.2
0.0
0.2
0.4
0.4
0.2
0.0
−0.2
−0.4
−0.6

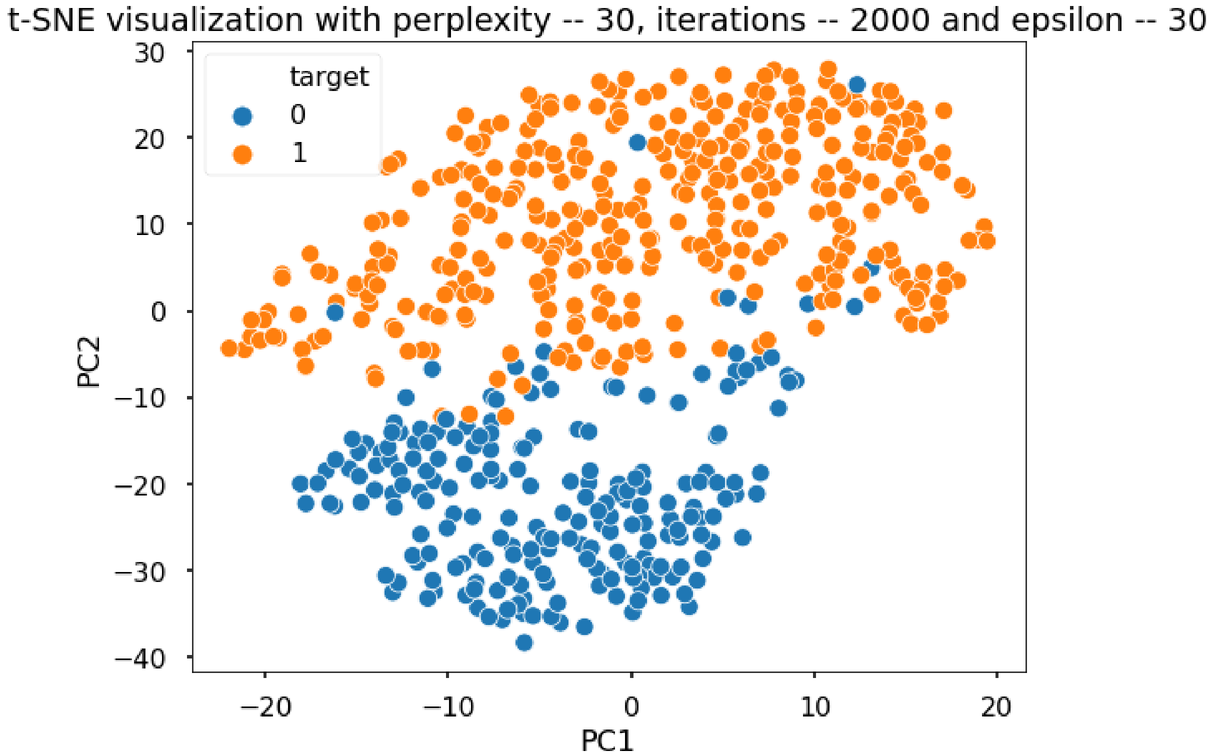

t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 30
target
0
1
PC1
PC2
−20
−10
0
10
20
30
15
10
5
0
−5
−10
−15

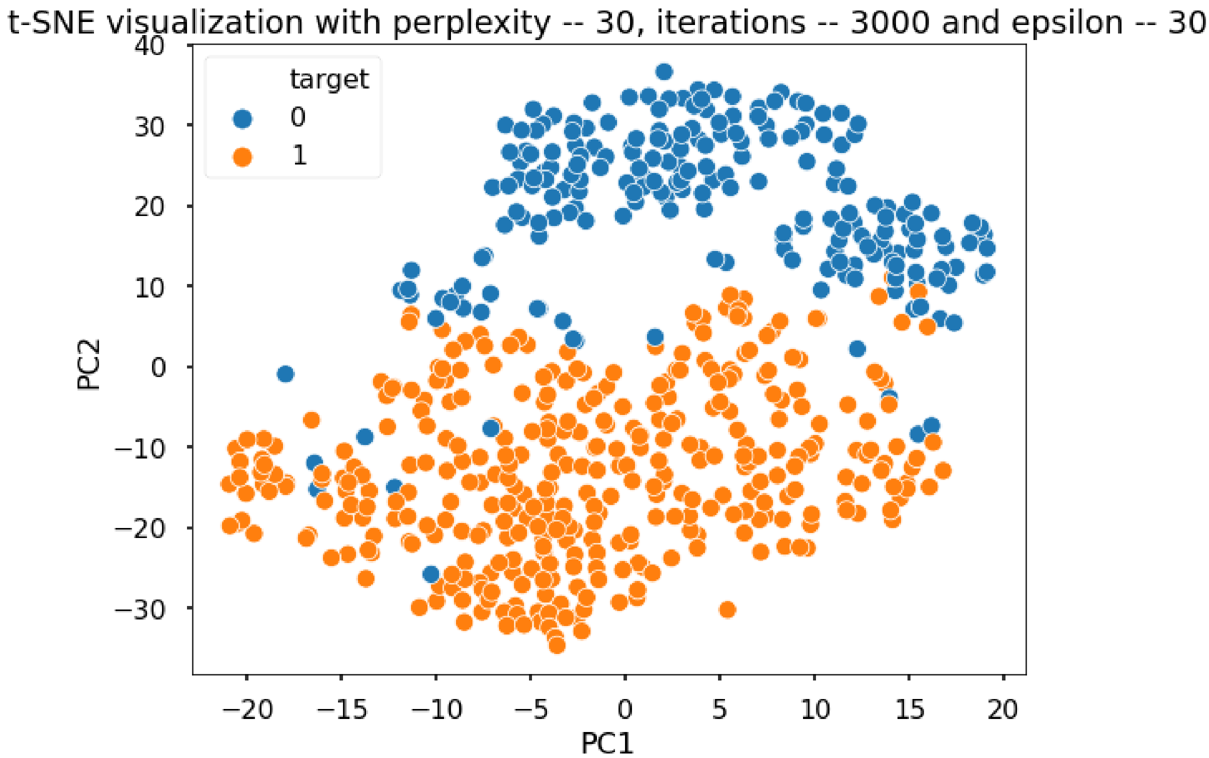
t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 30



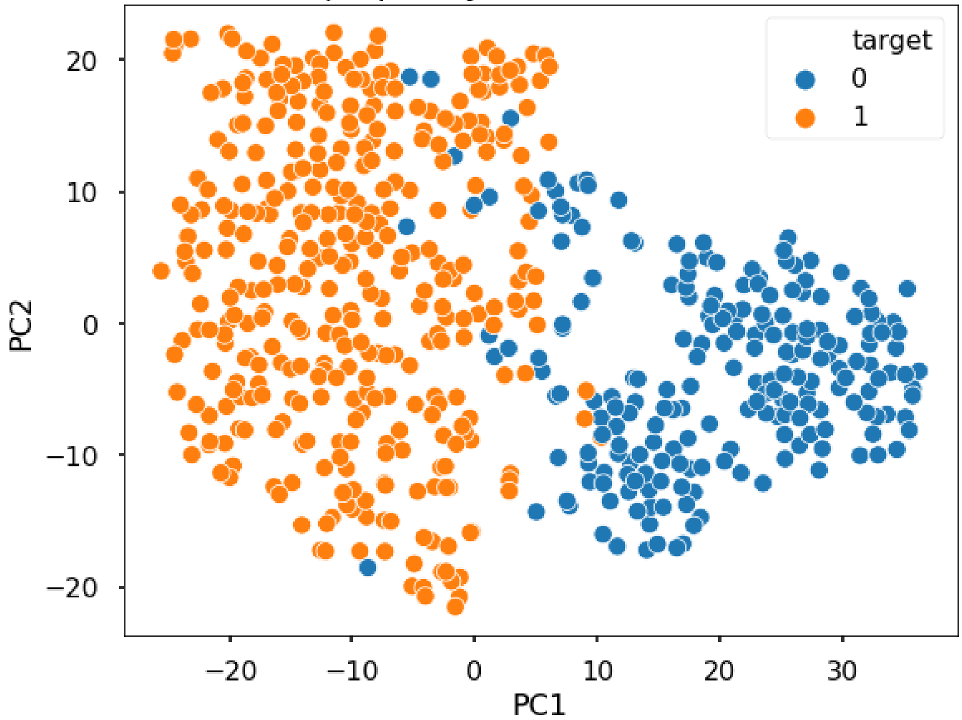
t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 30



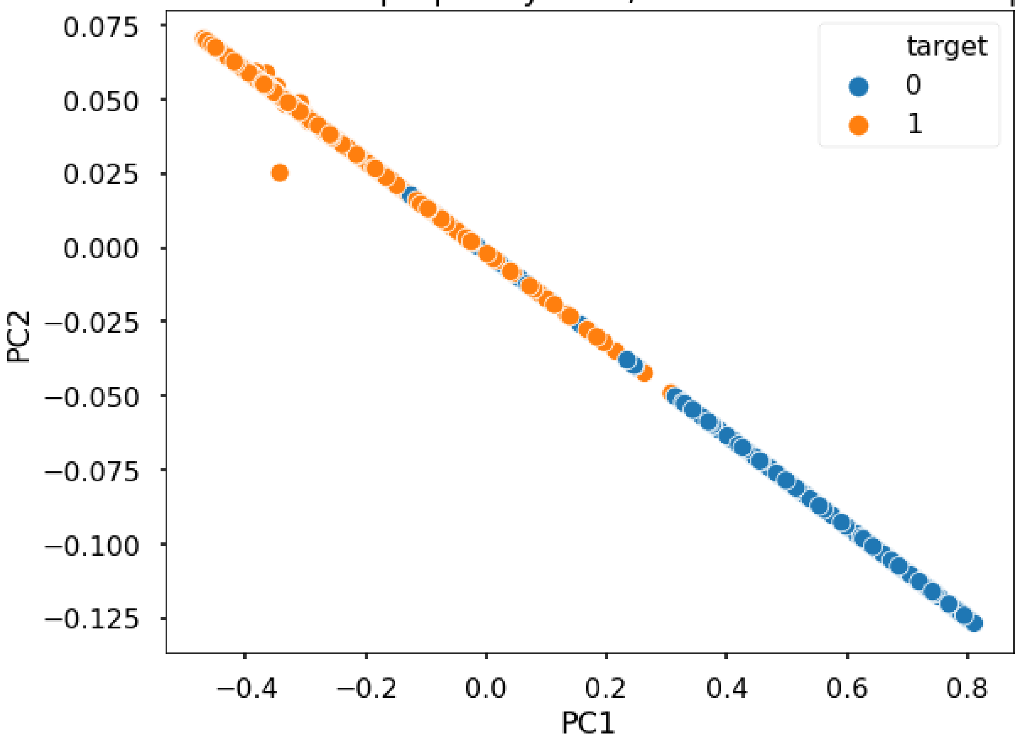

t-SNE visualization with perplexity -- 30, iterations -- 2000 and epsilon -- 30
target
0
1
PC2
PC1
30
20
10
0
−10
−20
−30
−40
−20
−10
0
10
20

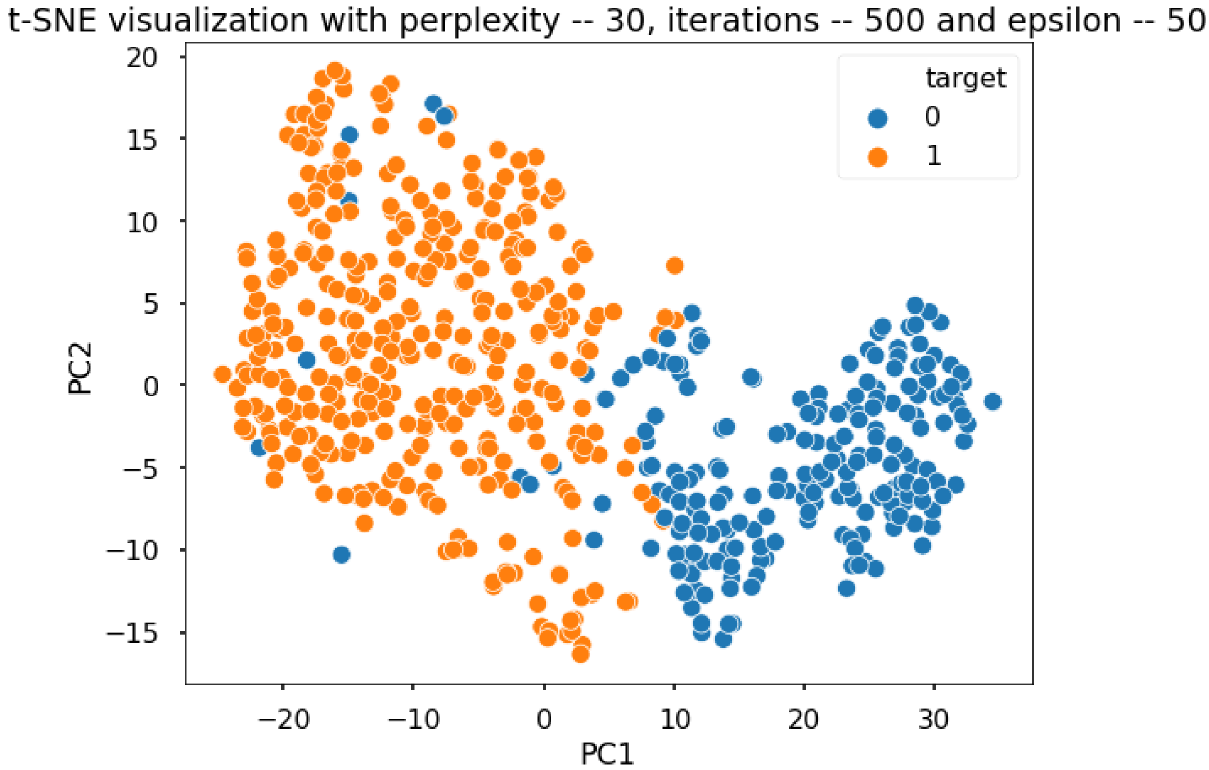

t-SNE visualization with perplexity -- 30, iterations -- 3000 and epsilon -- 30
target
0
1
PC2
PC1
40
30
20
10
0
−10
−20
−30
−20
−15
−10
−5
0
5
10
15
20

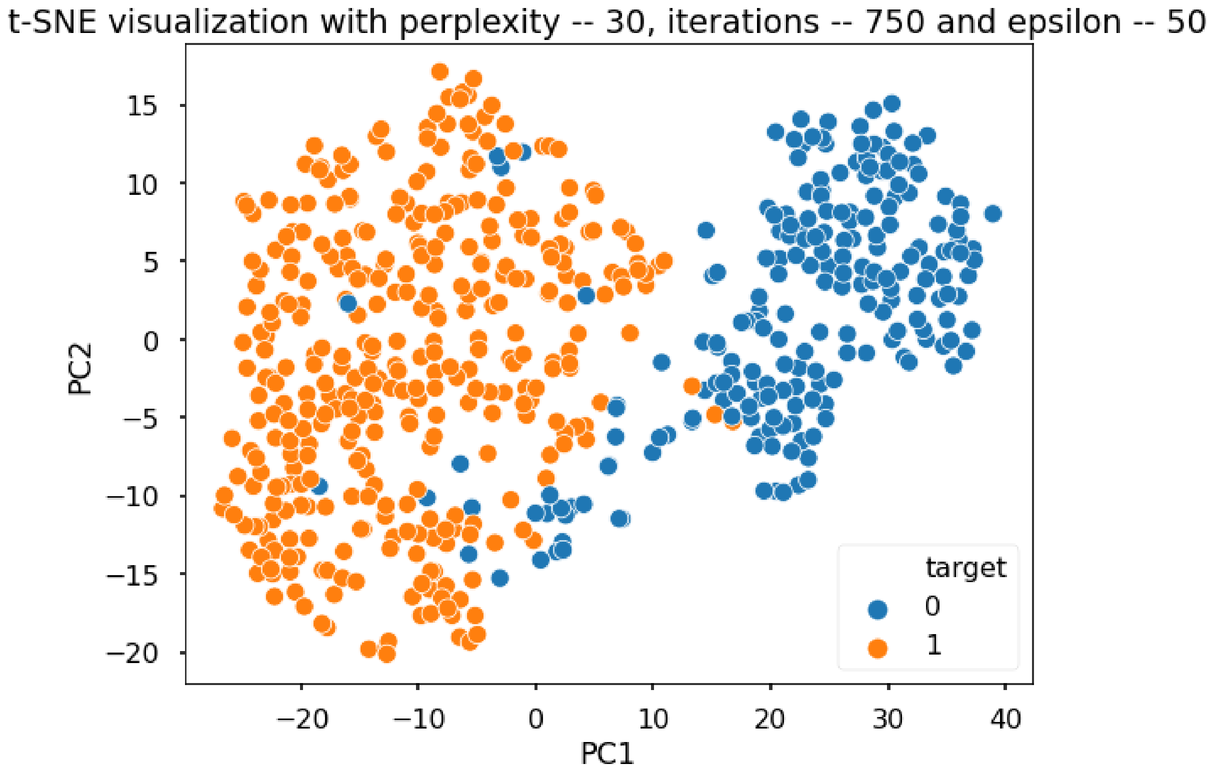
t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 30

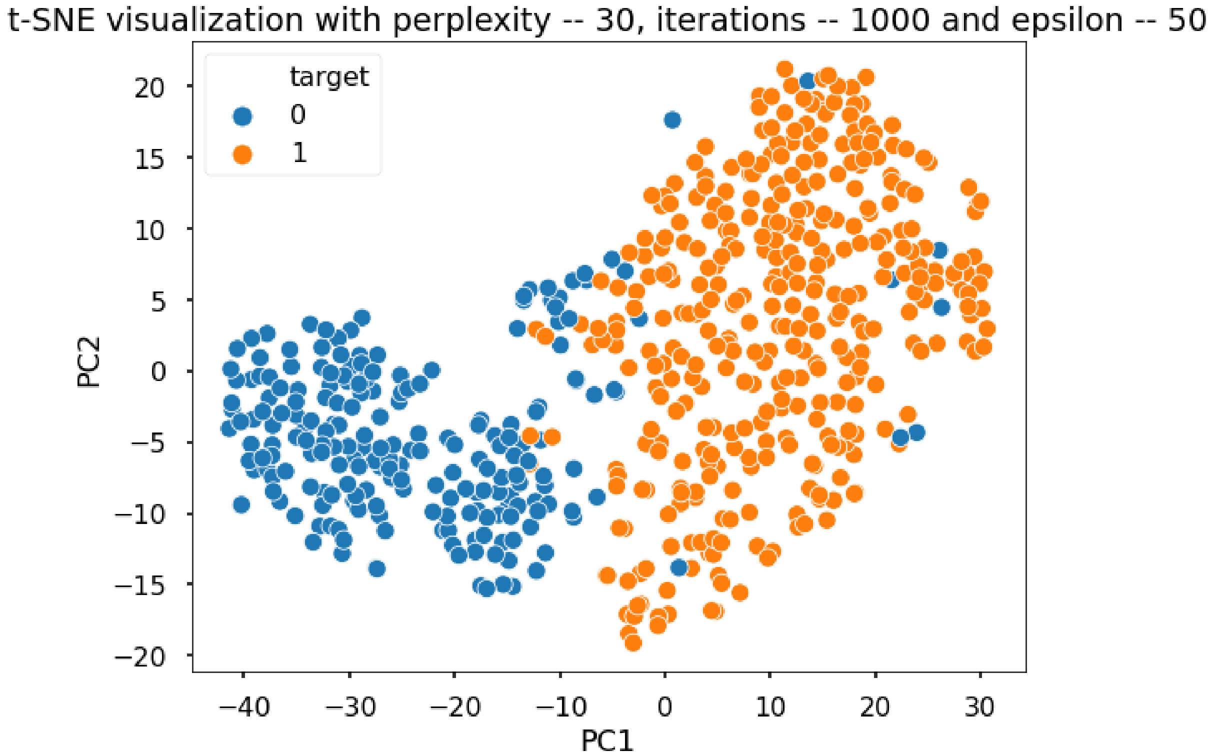


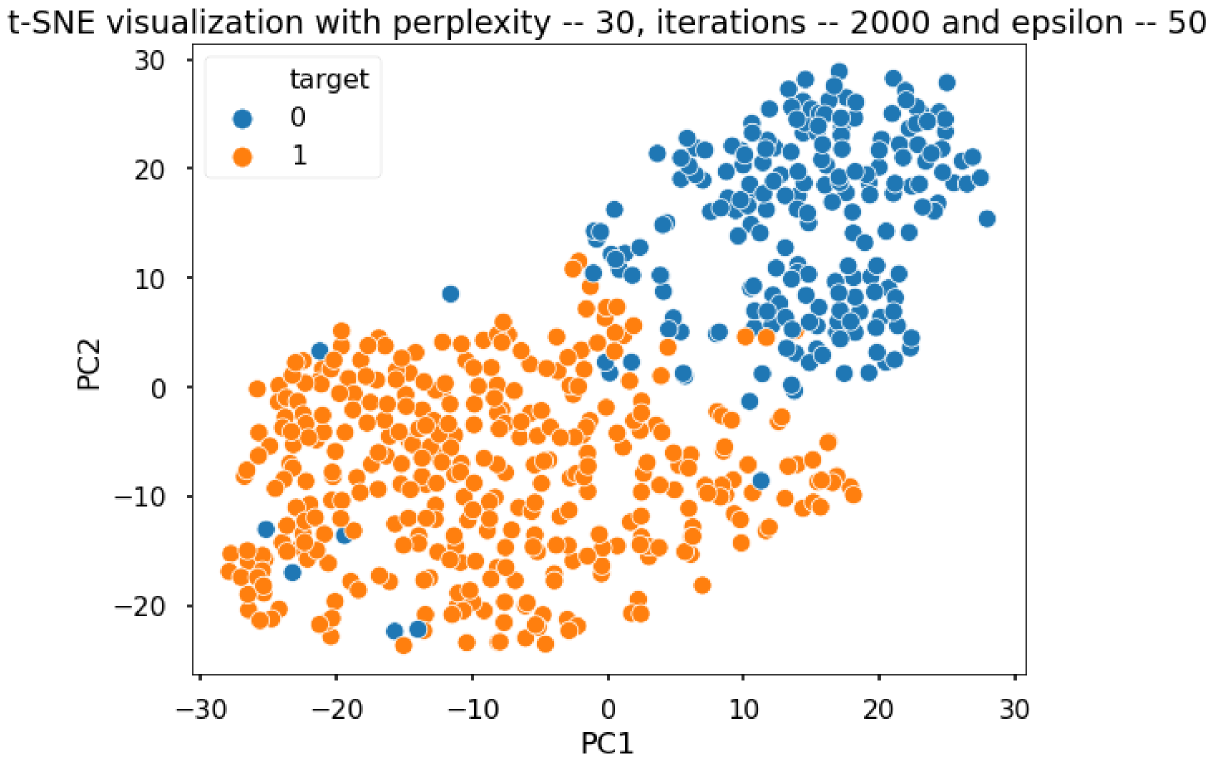
t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 50

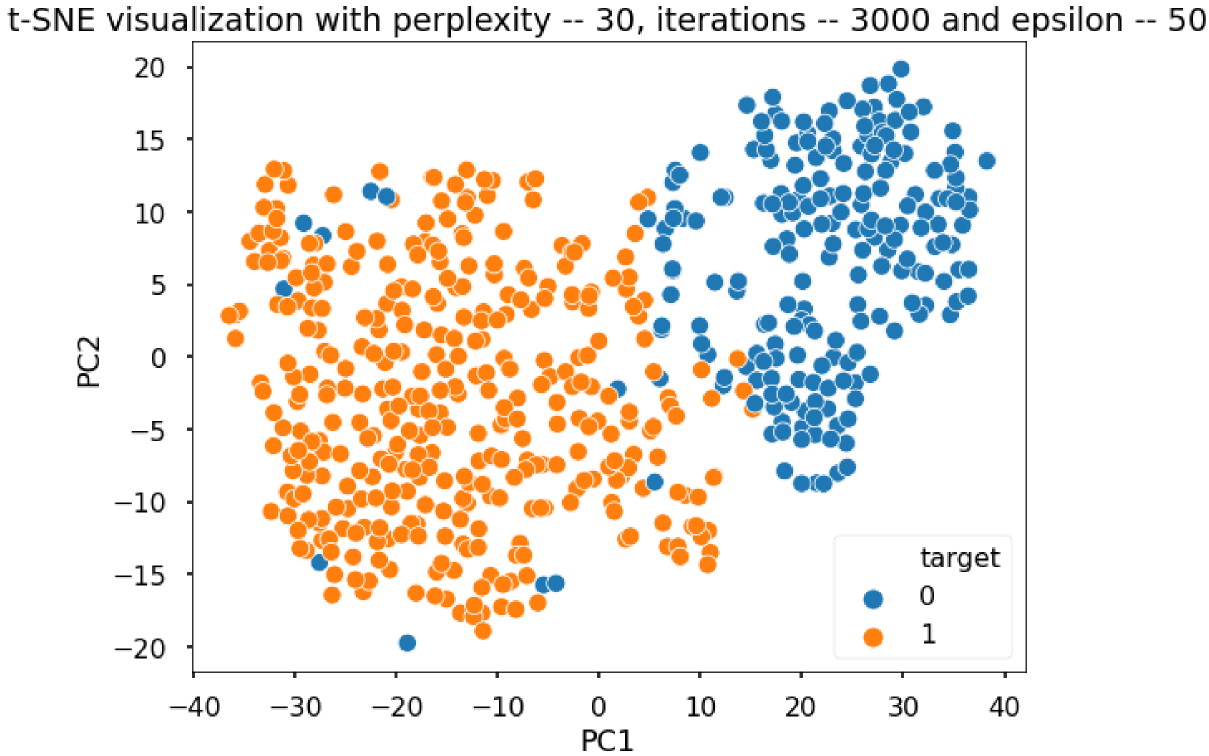


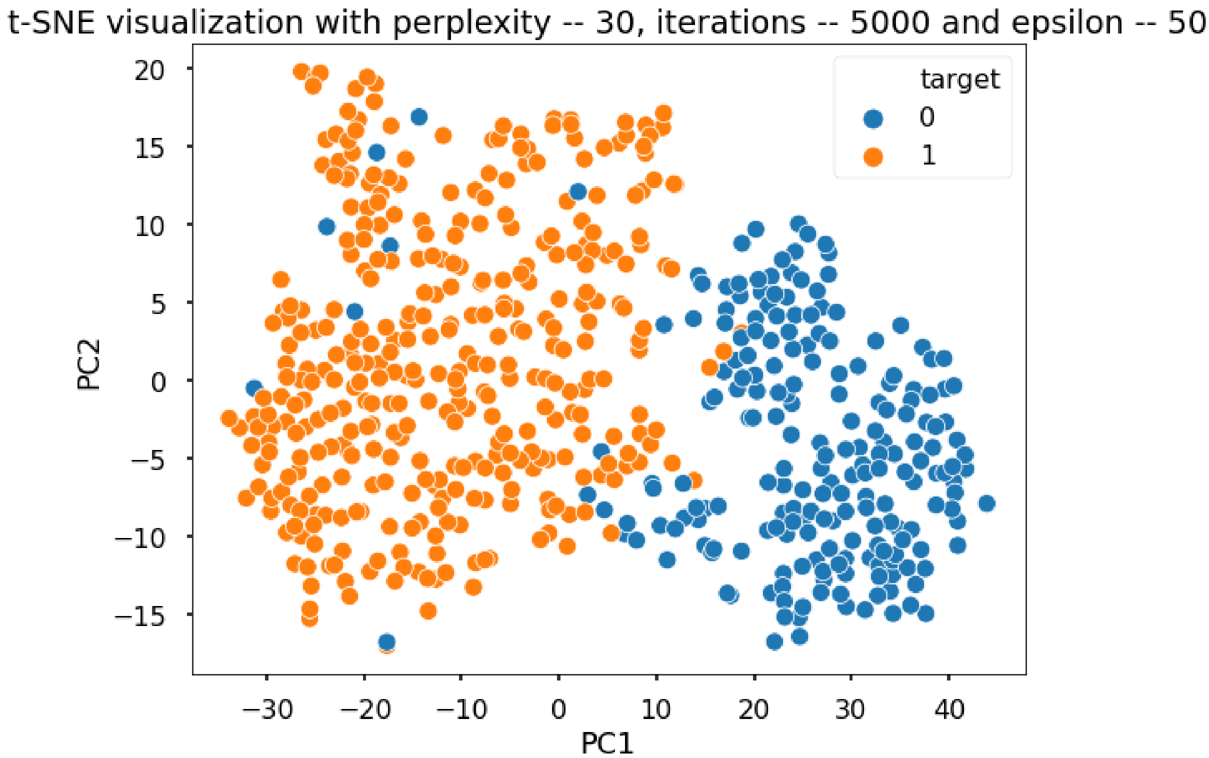

t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 50
target
0
1
PC2
PC1
20
15
10
5
0
−5
−10
−15
−20
−10
0
10
20
30


t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 50
target
0
1
PC2
PC1
15
10
5
0
−5
−10
−15
−20
−20
−10
0
10
20
30
40

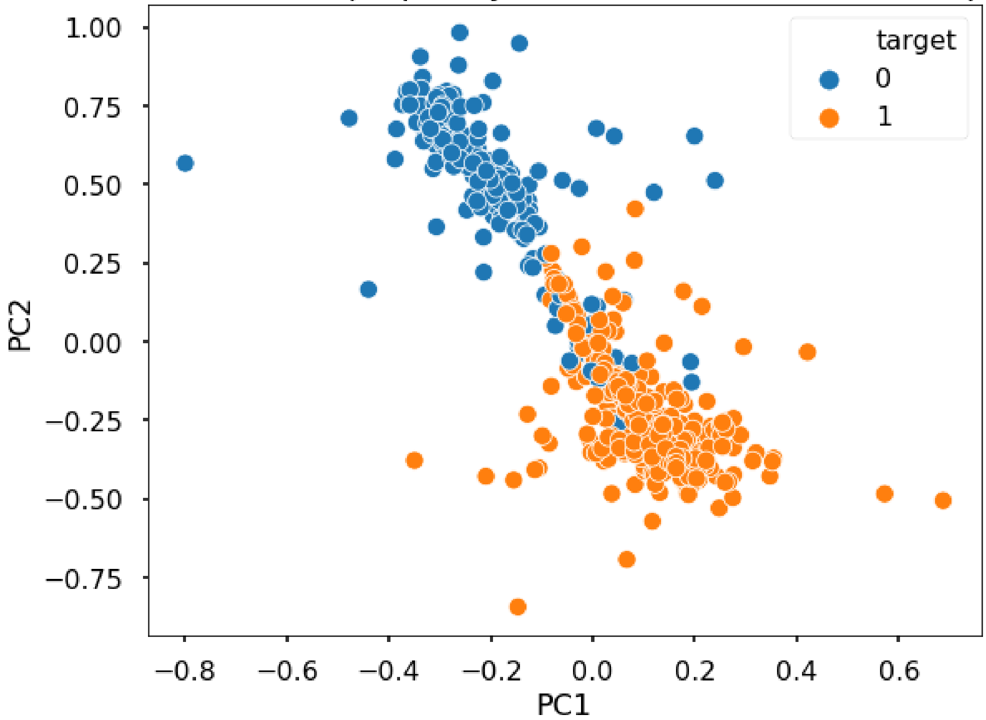

t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 50
target
0
1
PC2
PC1
20
15
10
5
0
−5
−10
−15
−20
−40
−30
−20
−10
0
10
20
30


t-SNE visualization with perplexity -- 30, iterations -- 2000 and epsilon -- 50
target
0
1
PC2
PC1
30
20
10
0
−10
−20
−30
−20
−10
0
10
20
30

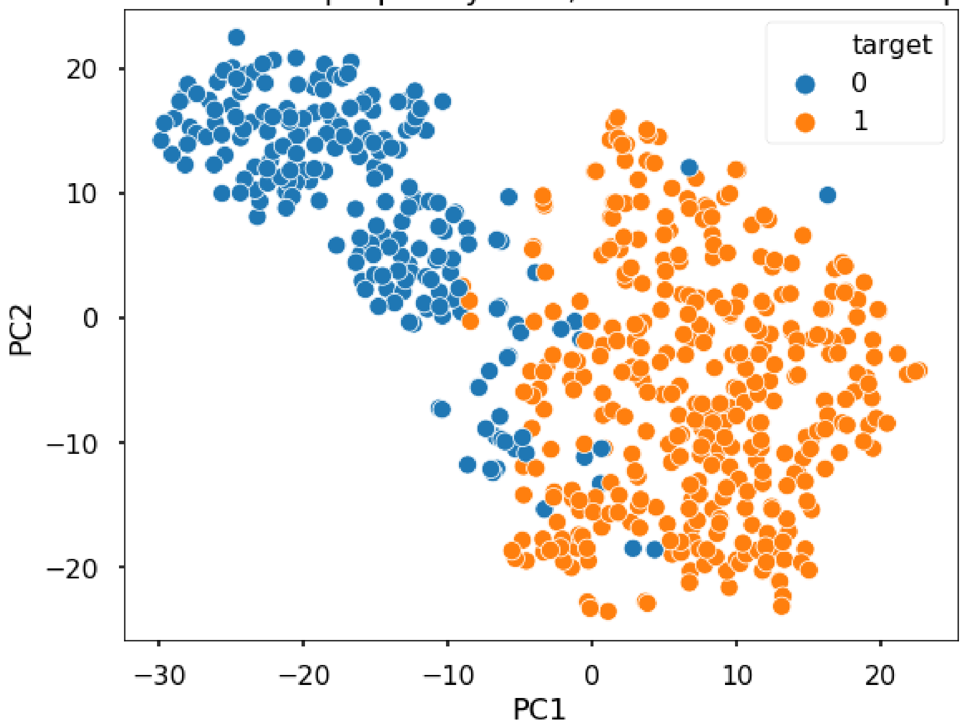

t-SNE visualization with perplexity -- 30, iterations -- 3000 and epsilon -- 50
PC2
PC1
target
0
1
20
15
10
5
0
−5
−10
−15
−20
−40
−30
−20
−10
0
10
20
30
40

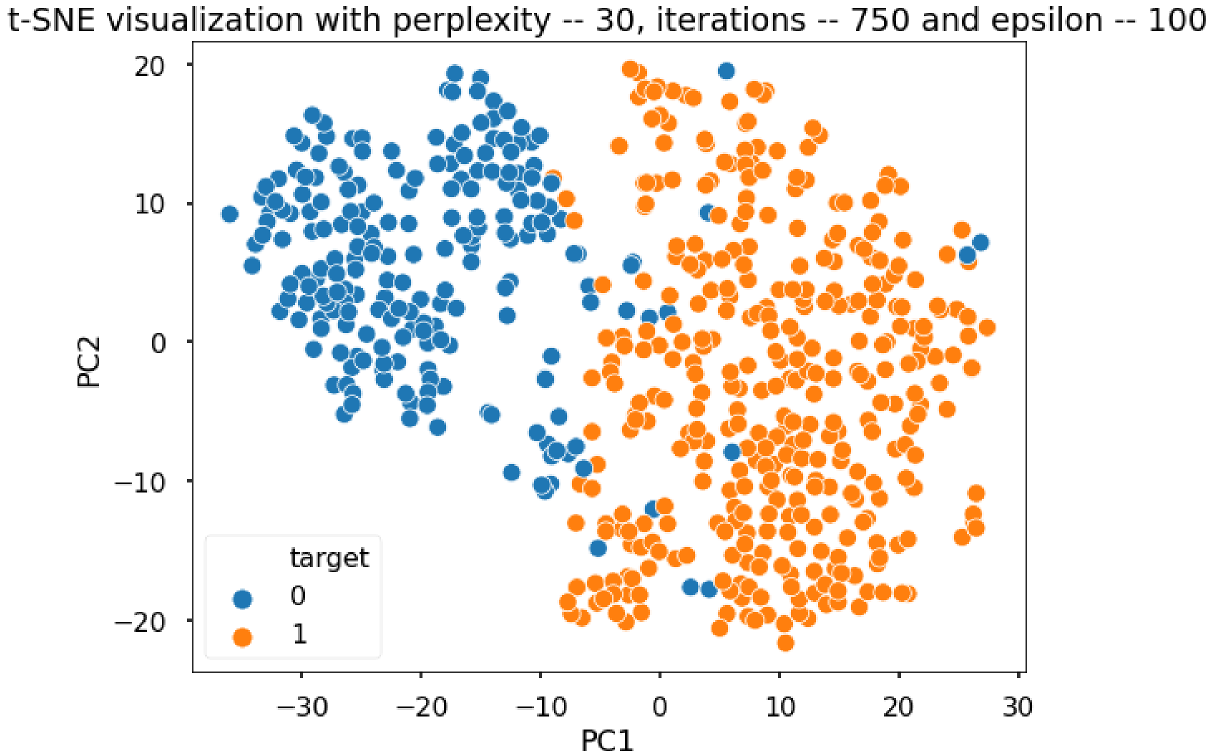

t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 50
PC2
PC1
target
0
1
20
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10
20
30
40

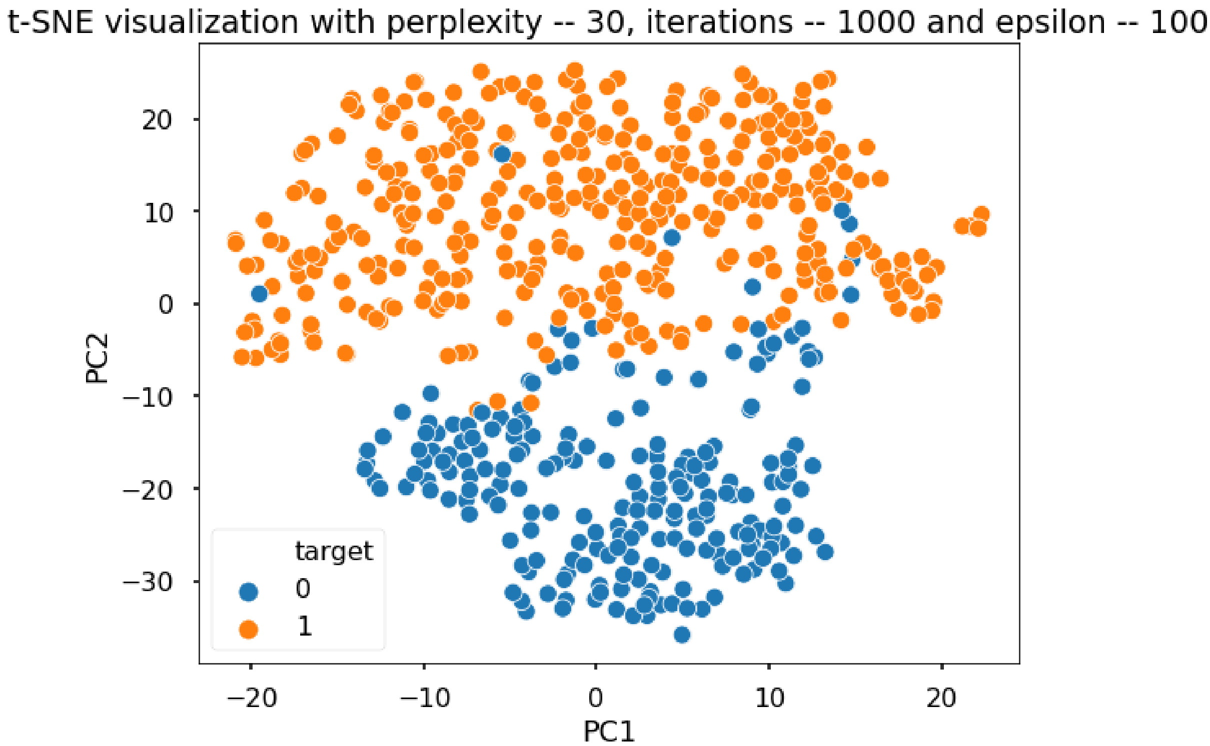
t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 100

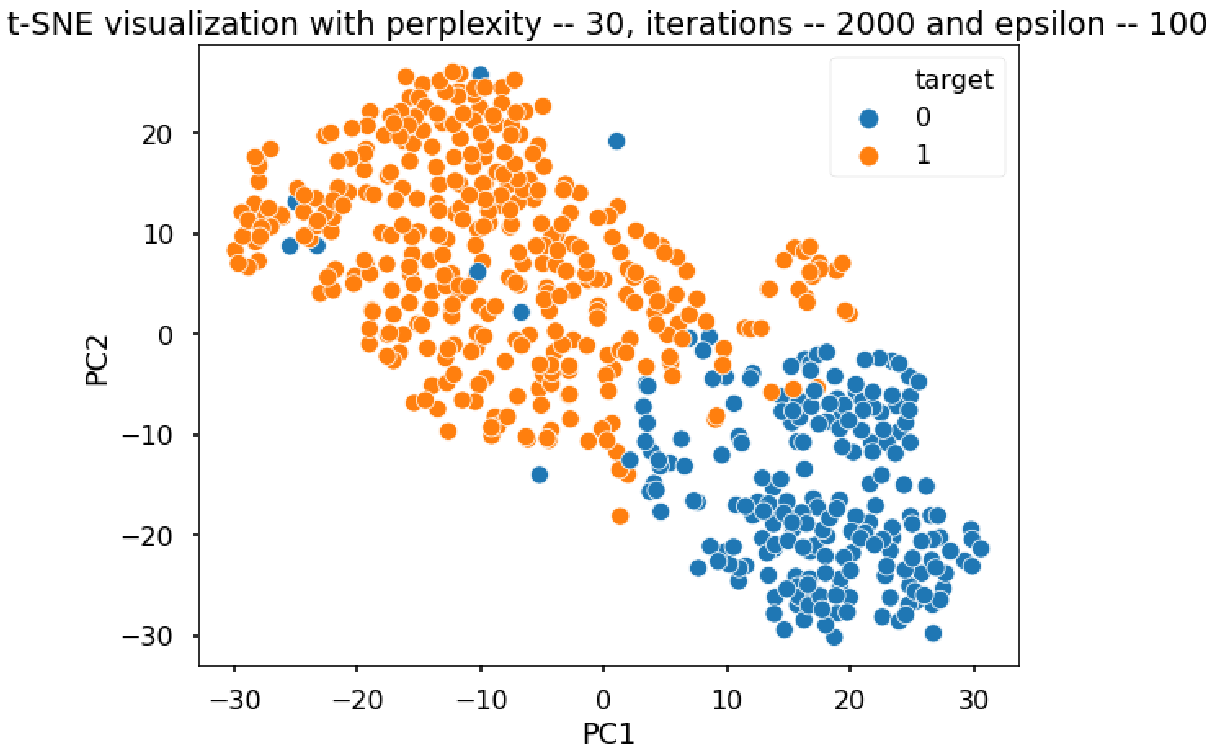


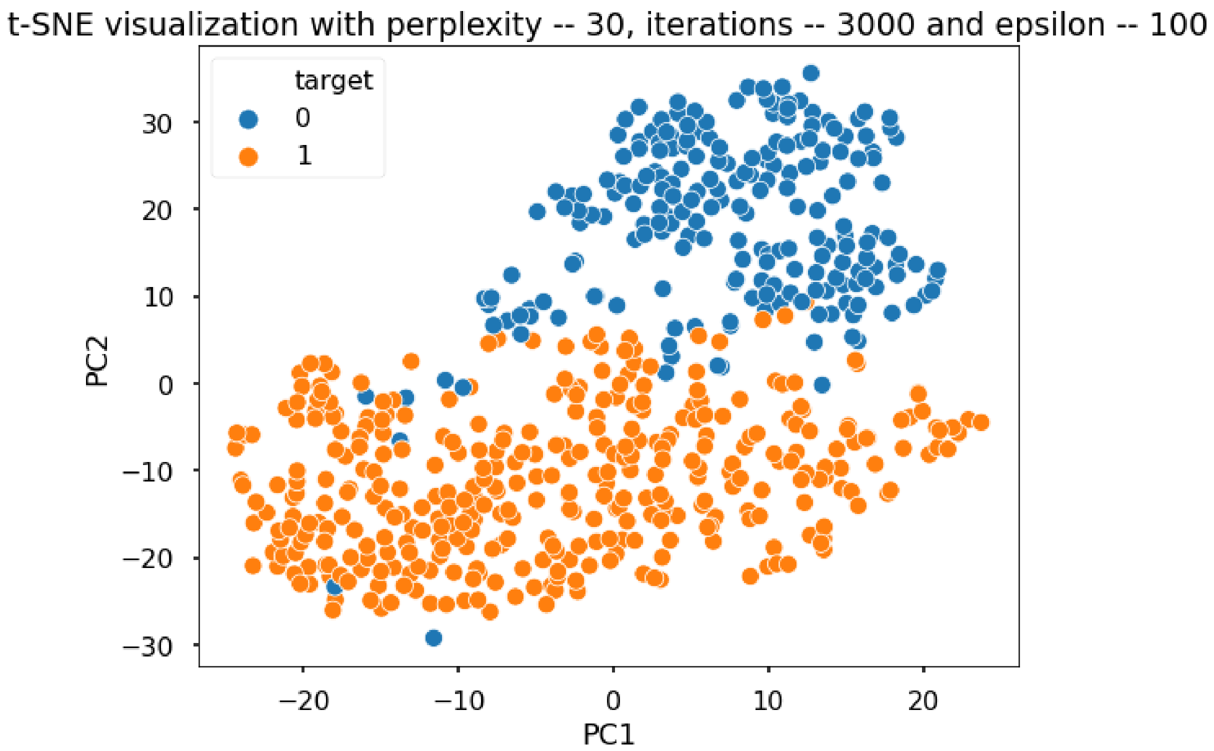
t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 100



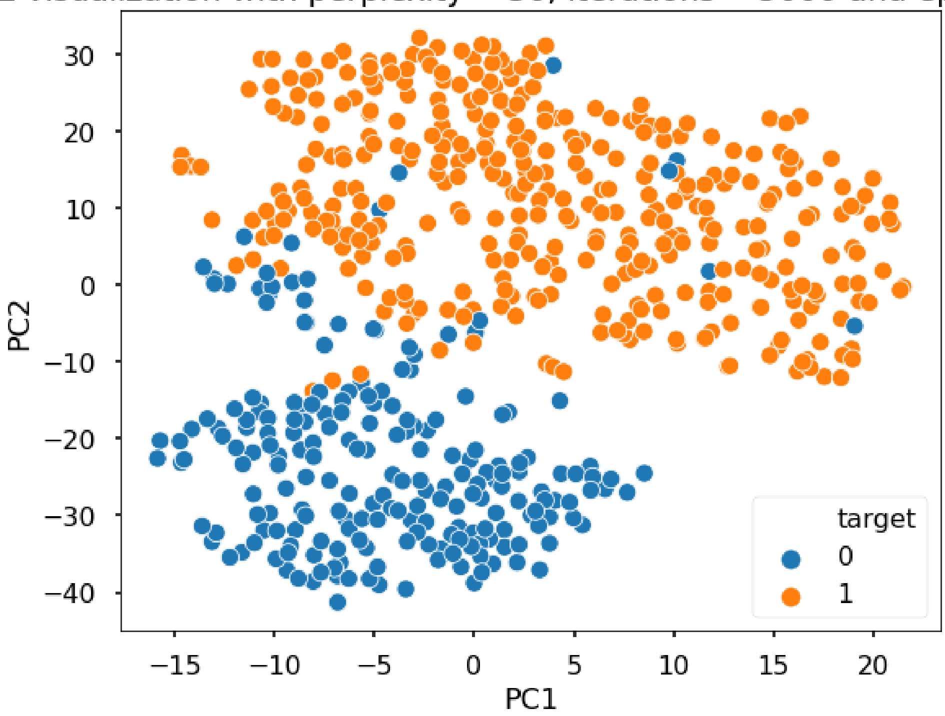

t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 100
target
0
1
PC1
PC2


t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 100
target
0
1
PC1
PC2

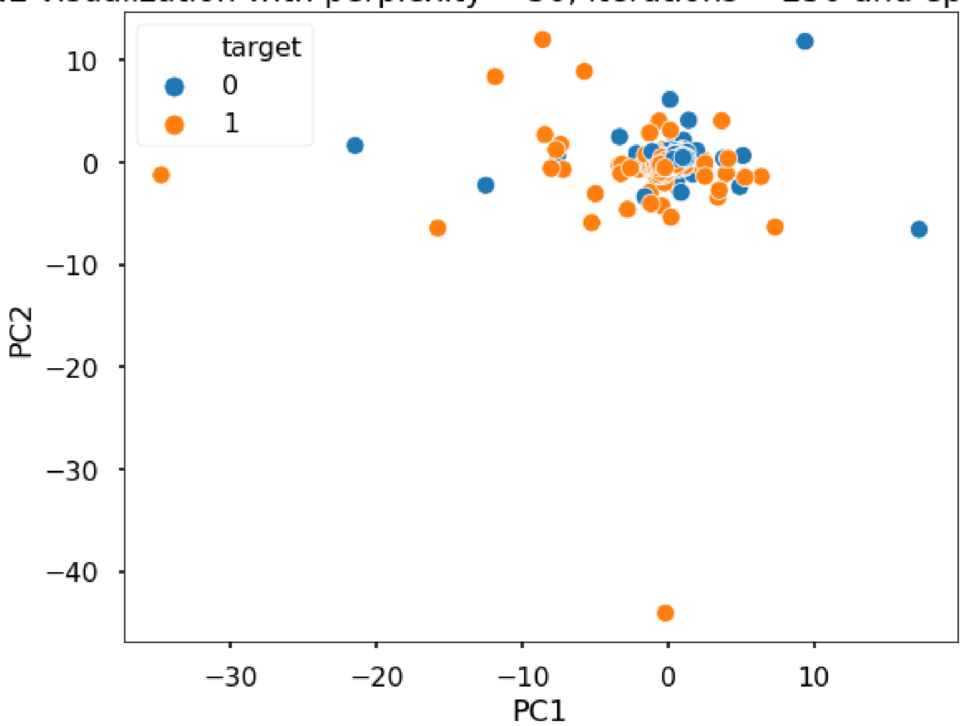

t-SNE visualization with perplexity -- 30, iterations -- 2000 and epsilon -- 100
target
0
1
PC2
PC1
20
10
0
−10
−20
−30
−30
−20
−10
0
10
20
30

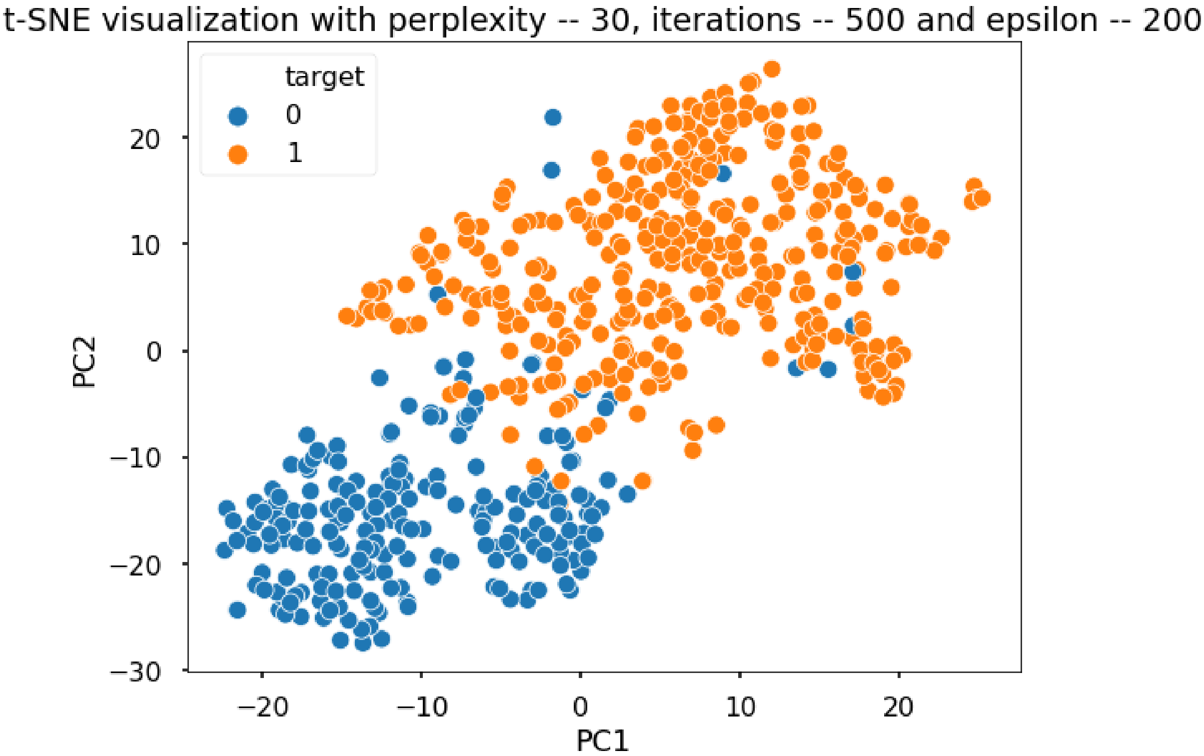

t-SNE visualization with perplexity -- 30, iterations -- 3000 and epsilon -- 100
target
0
1
PC2
PC1
30
20
10
0
−10
−20
−30
−20
−10
0
10
20

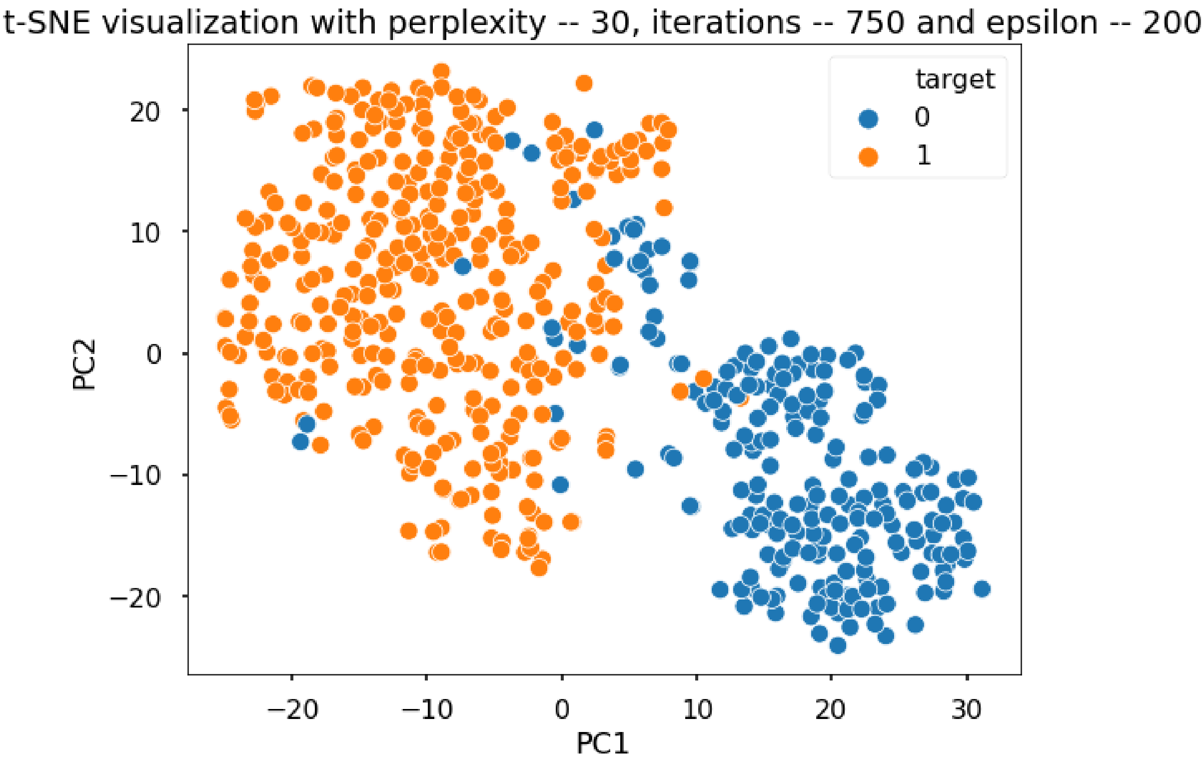
t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 100

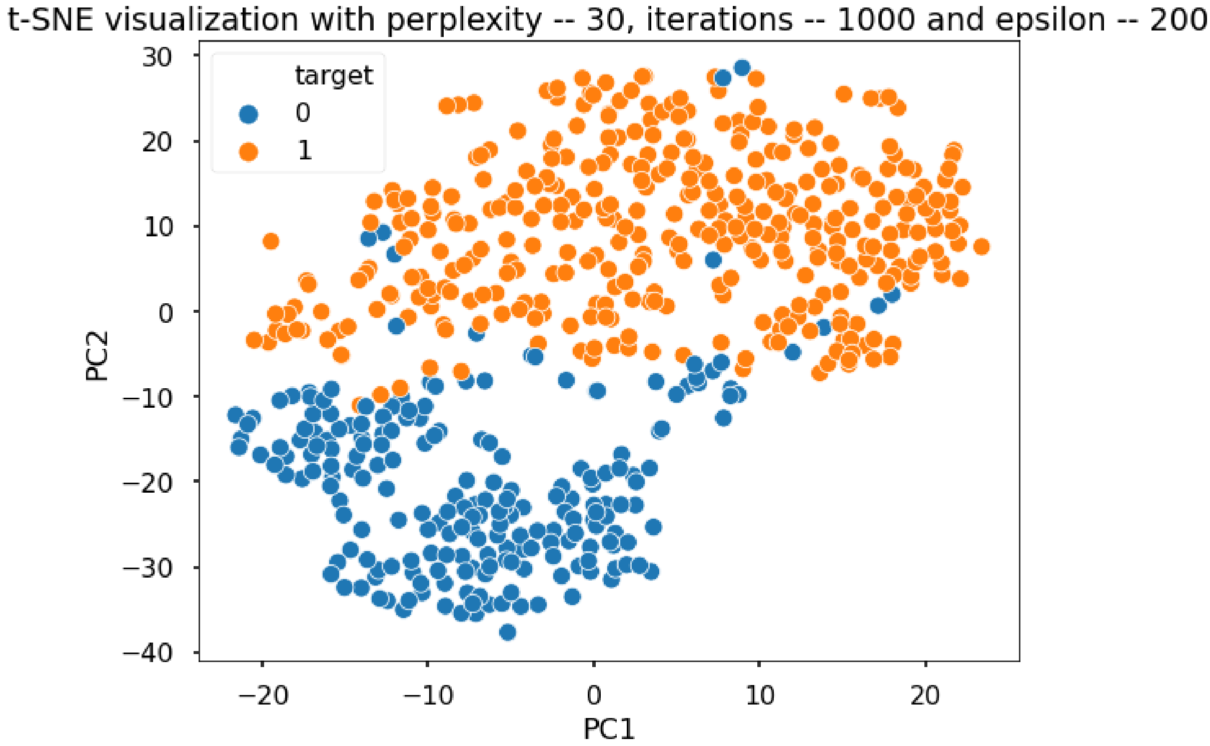


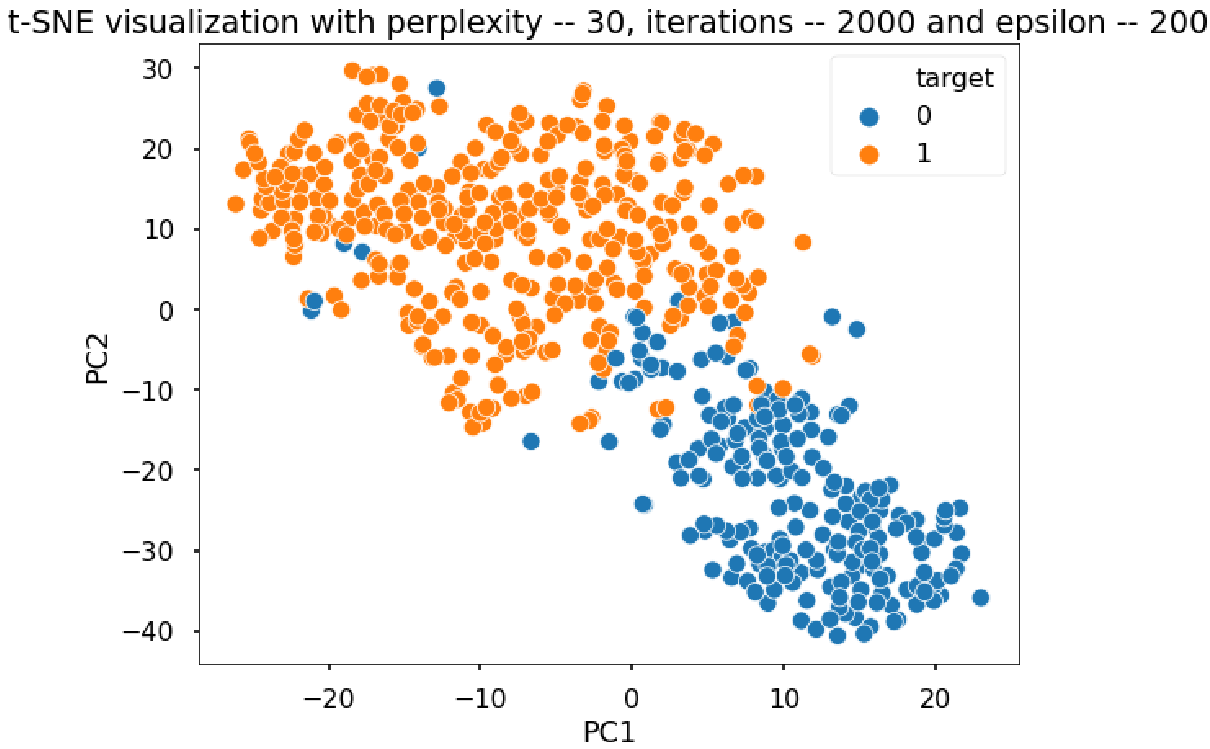
t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 200

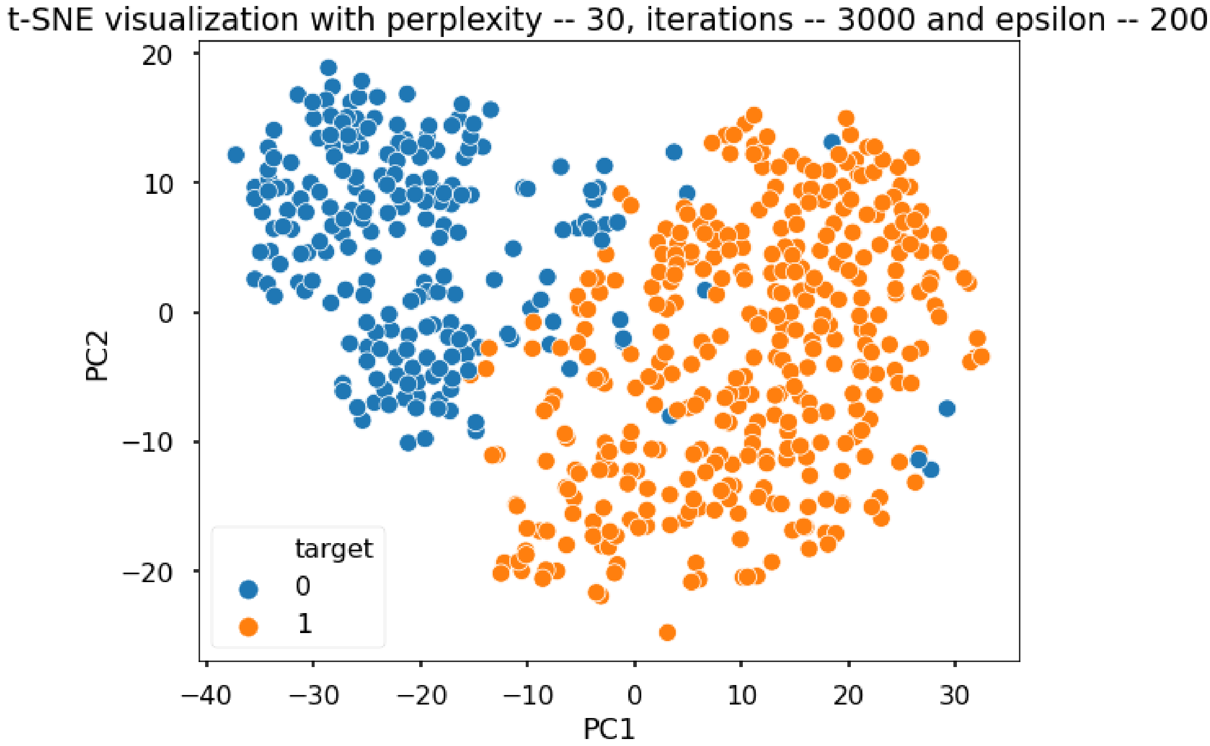



t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 200
target
0
1
20
10
0
−10
−20
−30
PC2
−20
−10
0
10
20
PC1

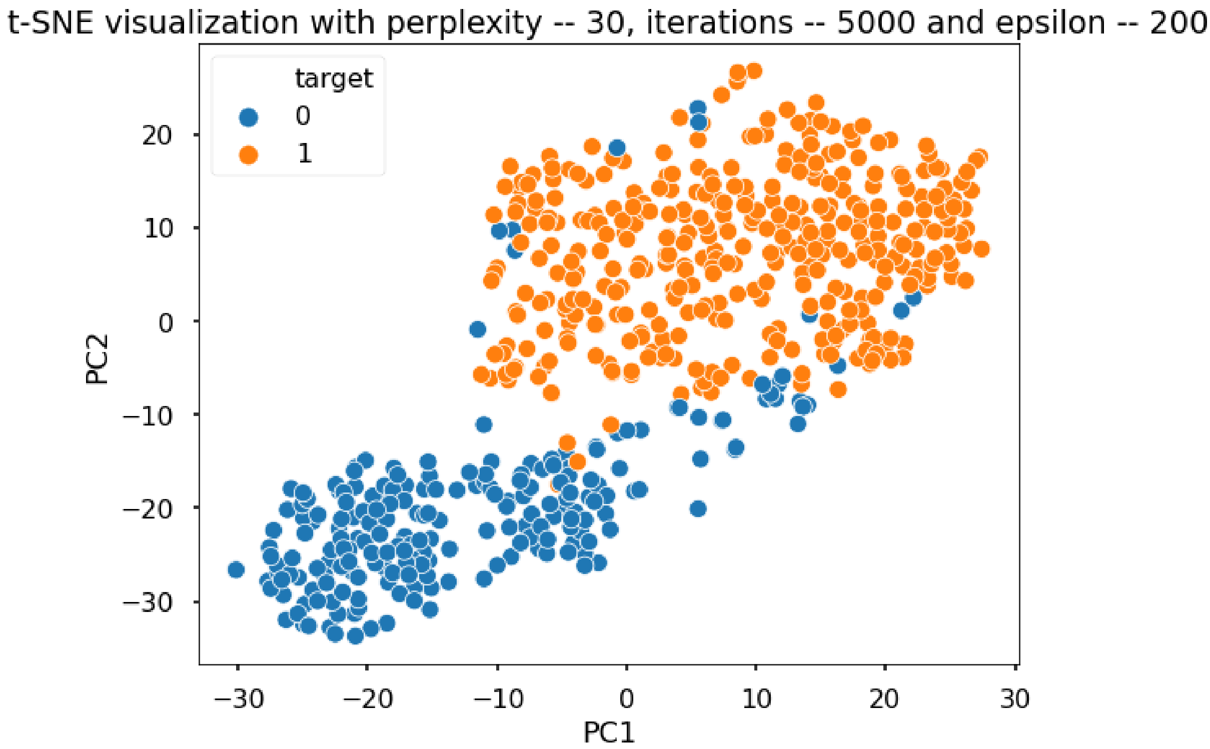

t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 200
target
0
1
20
10
0
−10
−20
PC2
−20
−10
0
10
20
30
PC1


t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 200
target
0
1
PC2
PC1
30
20
10
0
−10
−20
−30
−40
−20
−10
0
10
20

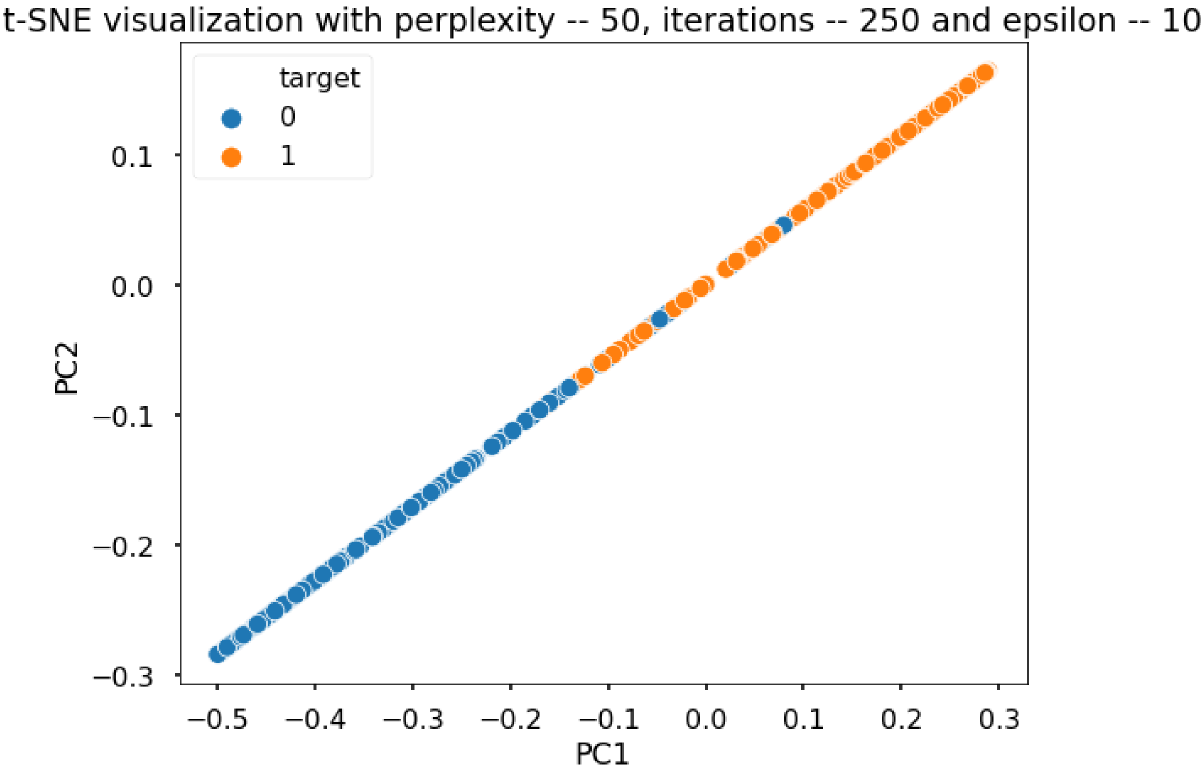

t-SNE visualization with perplexity -- 30, iterations -- 2000 and epsilon -- 200
target
0
1
PC2
PC1
30
20
10
0
−10
−20
−30
−40
−20
−10
0
10
20

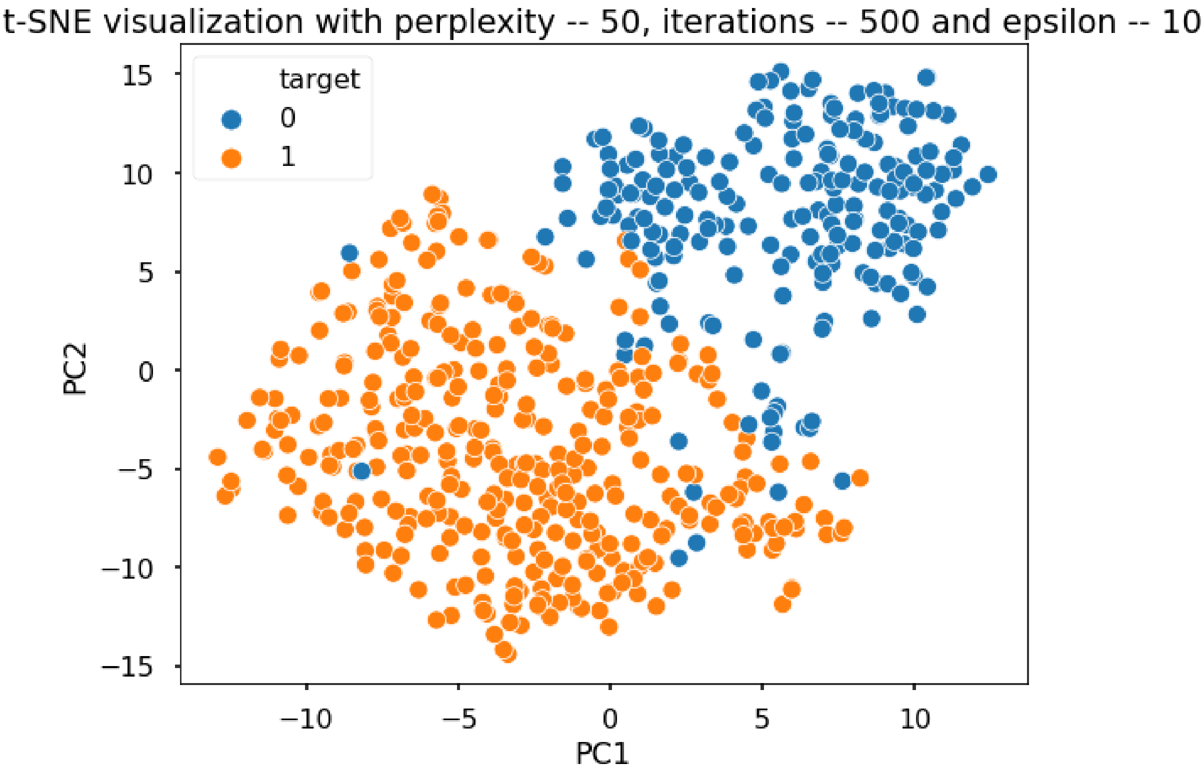


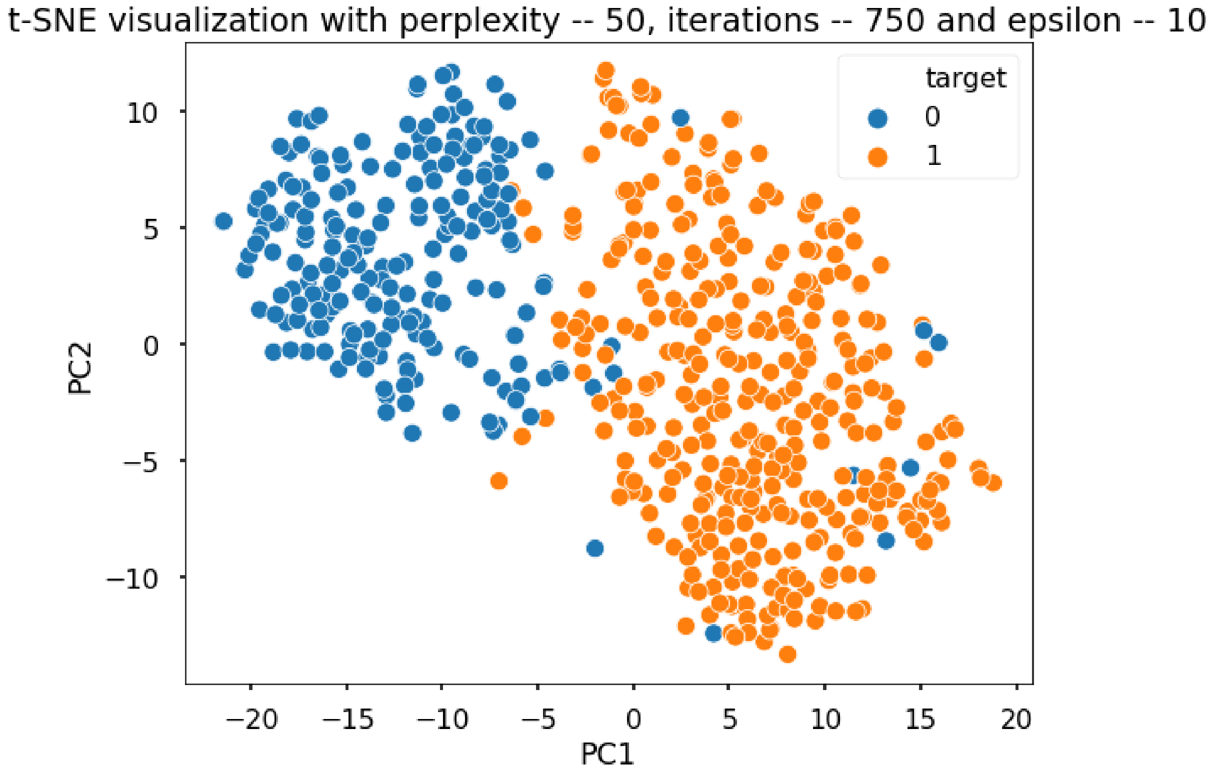
t-SNE visualization with perplexity -- 30, iterations -- 3000 and epsilon -- 200
PC2
PC1
target
0
1
20
10
0
−10
−20
−40
−30
−20
−10
0
10
20
30

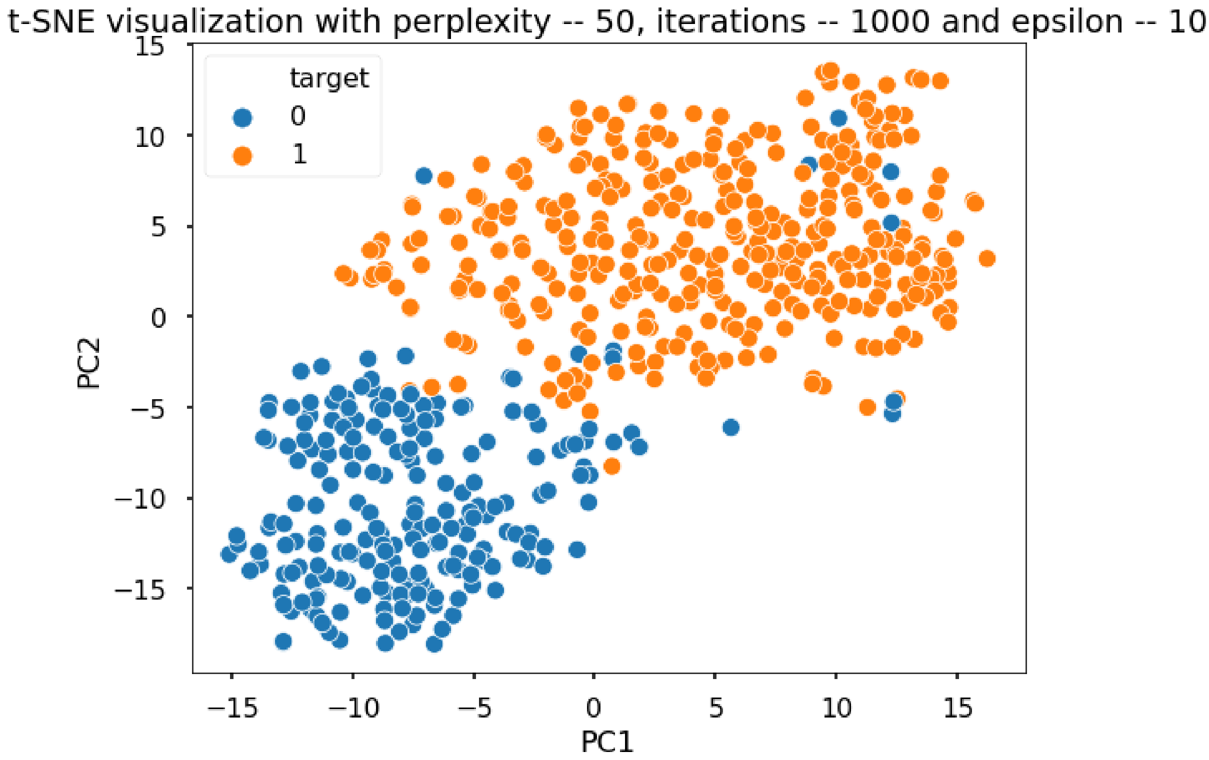


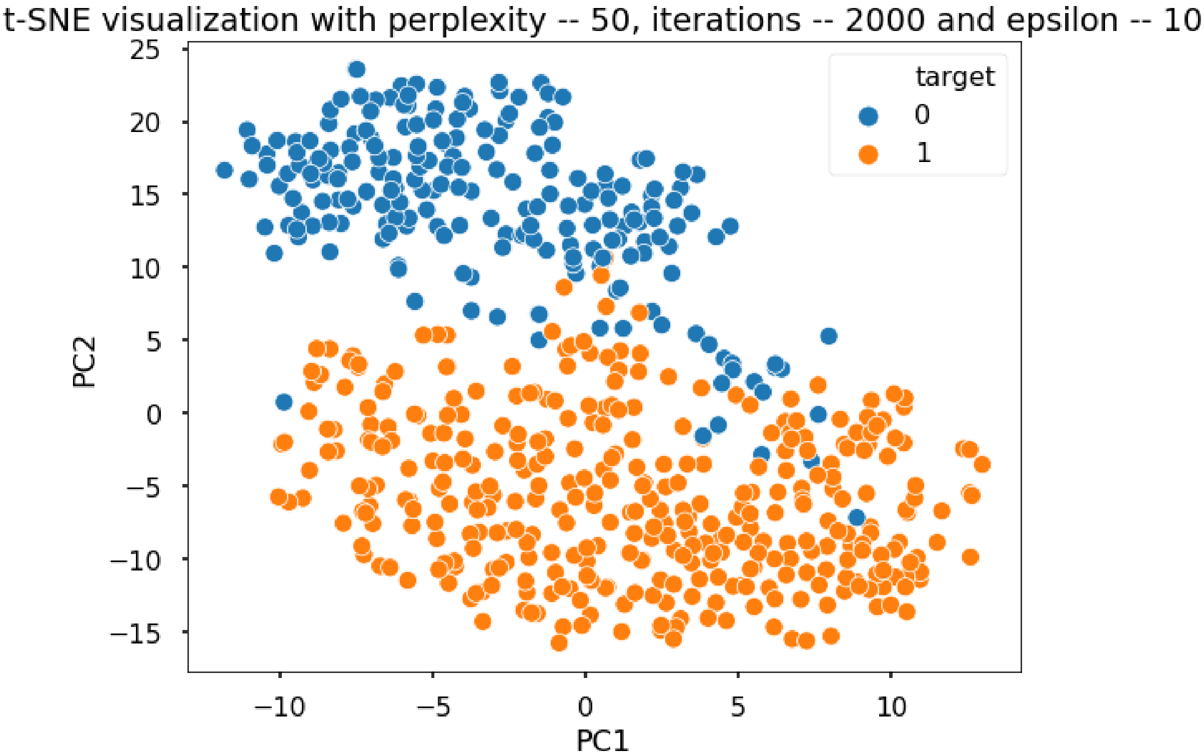
t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 200
PC2
PC1
target
0
1
20
10
0
−10
−20
−30
−30
−20
−10
0
10
20
30

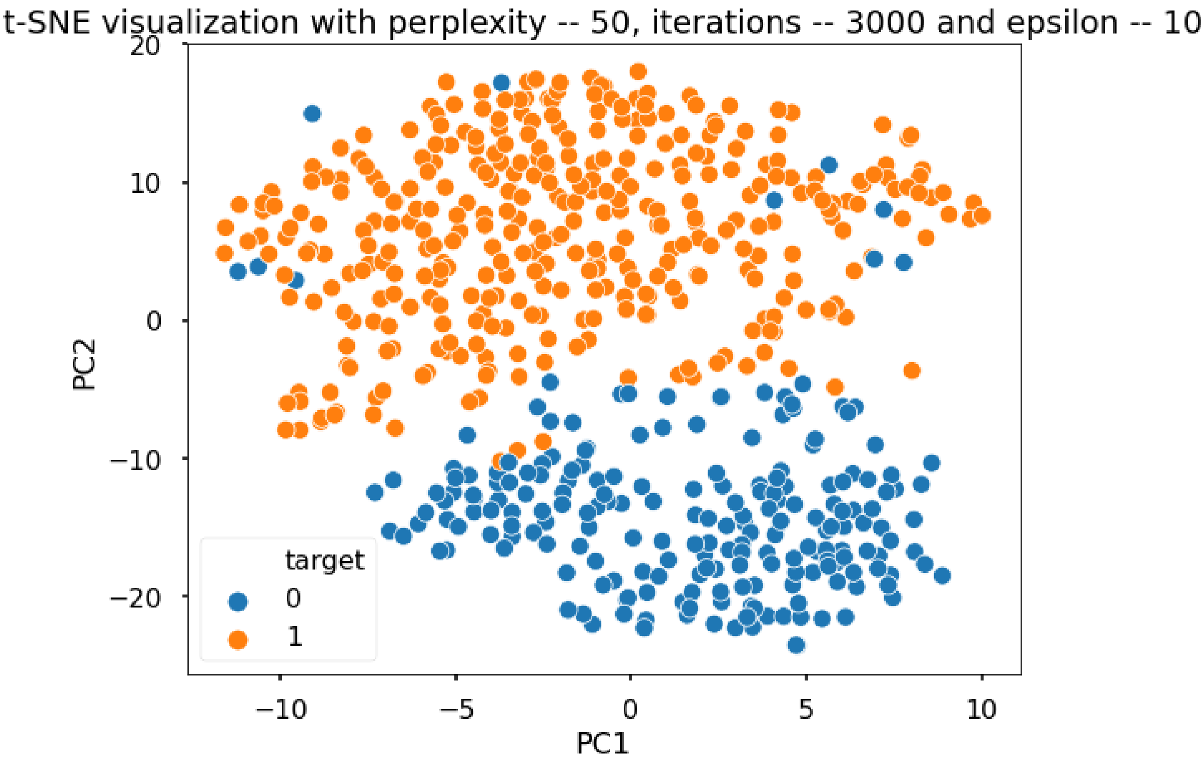

t-SNE visualization with perplexity -- 50, iterations -- 250 and epsilon -- 10
target
0
1
0.1
0.0
−0.1
−0.2
−0.3
PC2
−0.5
−0.4
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
PC1

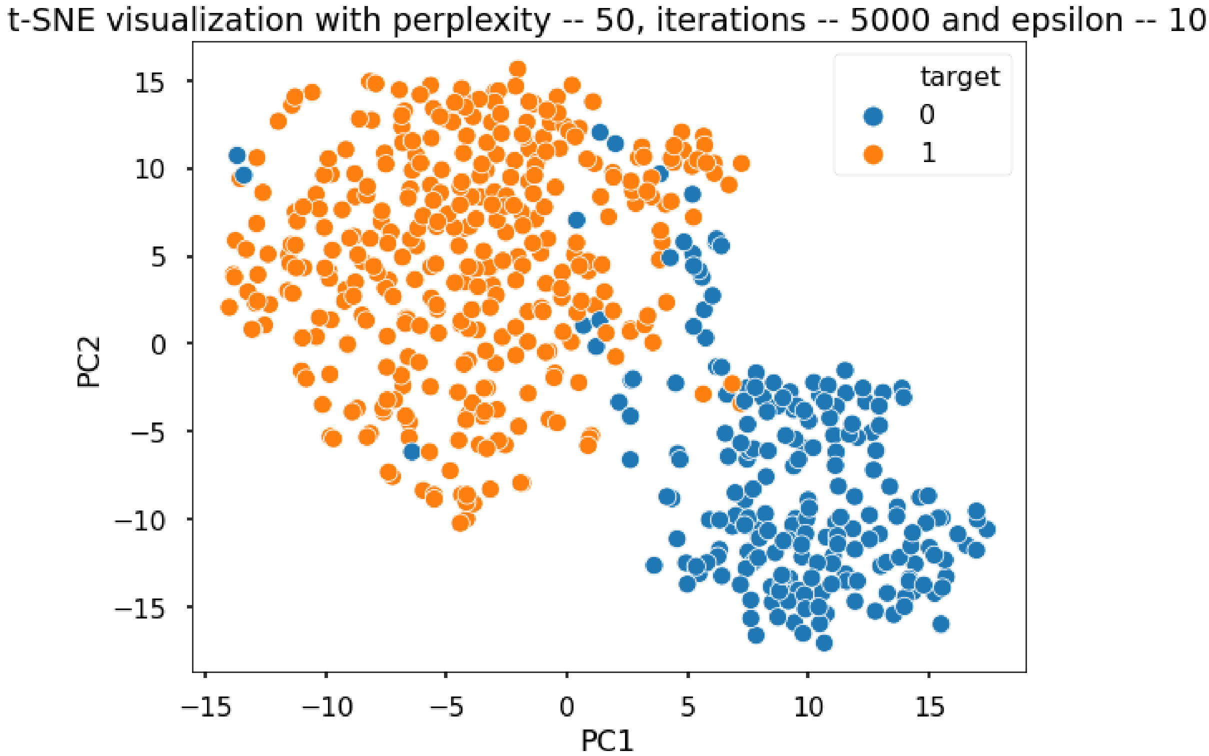

t-SNE visualization with perplexity -- 50, iterations -- 500 and epsilon -- 10
target
0
1
15
10
5
0
−5
−10
−15
PC2
−10
−5
0
5
10
PC1

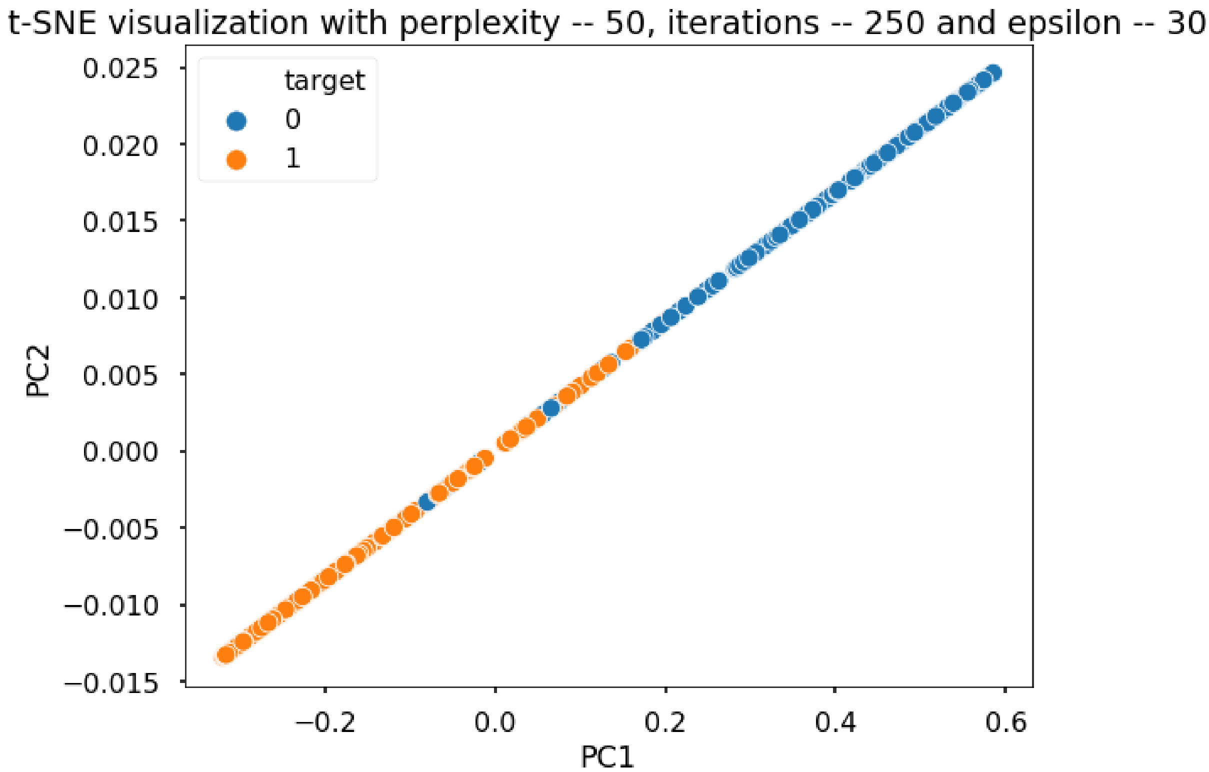

t-SNE visualization with perplexity -- 50, iterations -- 750 and epsilon -- 10
target
0
1
PC2
PC1
10
5
0
−5
−10
−20
−15
−10
−5
0
5
10
15
20

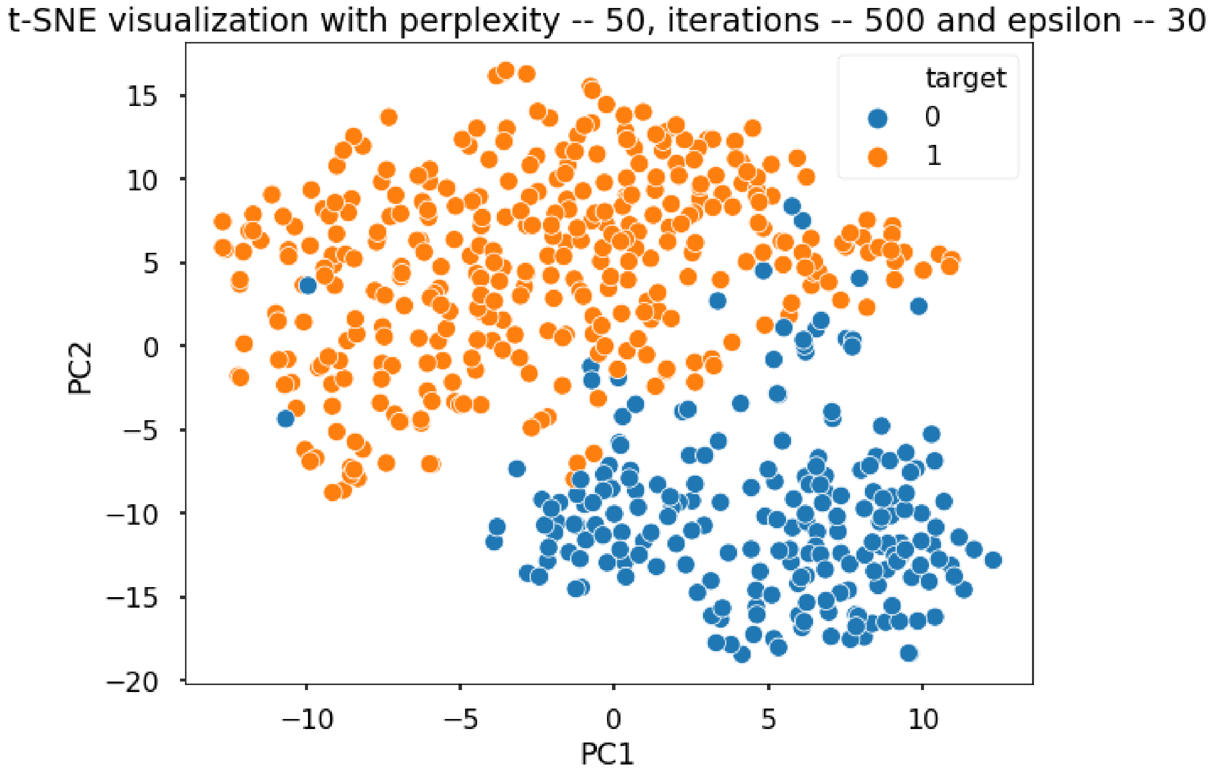

t-SNE visualization with perplexity -- 50, iterations -- 1000 and epsilon -- 10
target
0
1
PC2
PC1
15
10
5
0
−5
−10
−15
−15
−10
−5
0
5
10
15

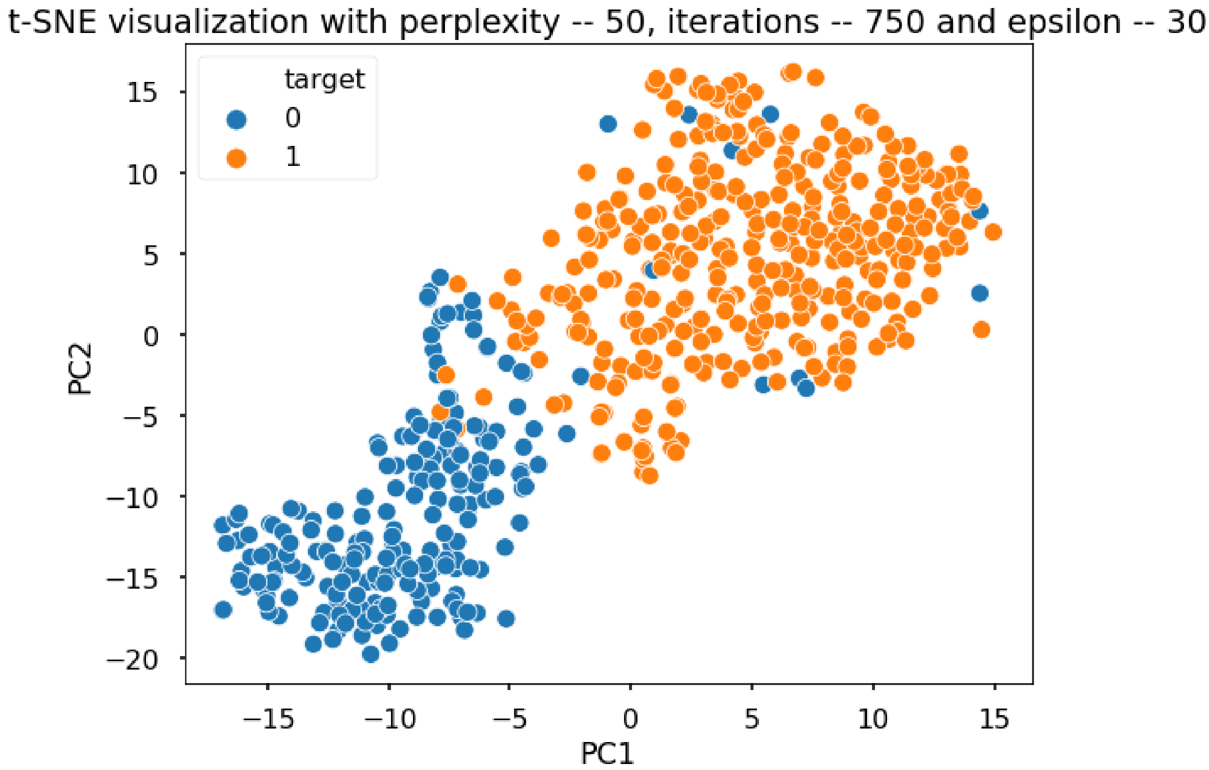

t-SNE visualization with perplexity -- 50, iterations -- 2000 and epsilon -- 10
target
0
1
PC2
PC1

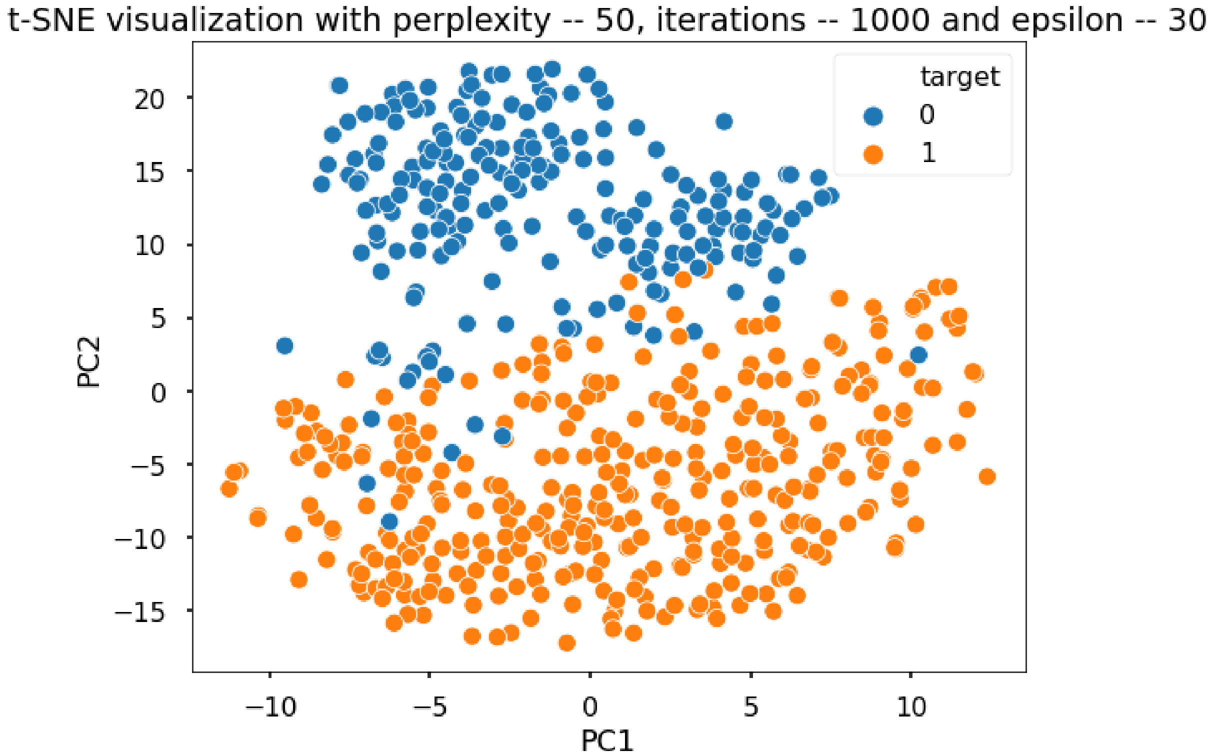

t-SNE visualization with perplexity -- 50, iterations -- 3000 and epsilon -- 10
target
0
1
PC2
PC1

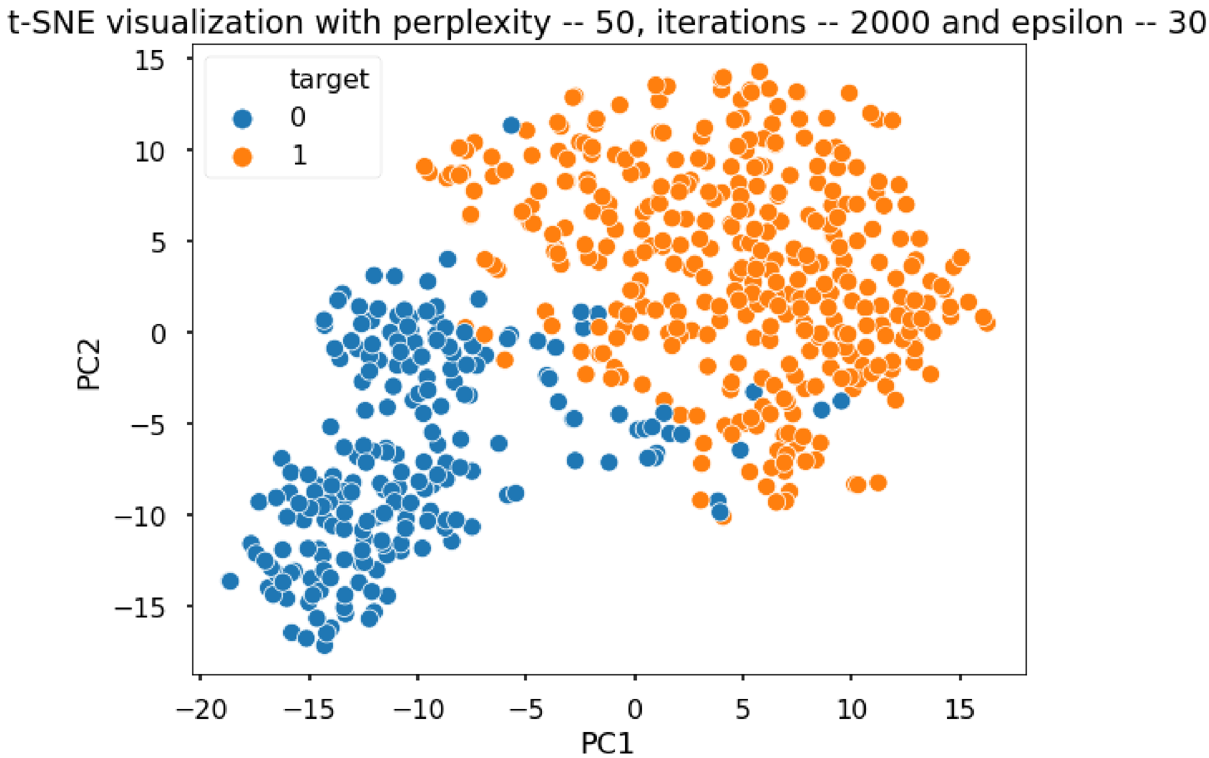

t-SNE visualization with perplexity -- 50, iterations -- 5000 and epsilon -- 10
target
0
1
PC2
PC1
15
10
5
0
−5
−10
−15
−15
−10
−5
0
5
10
15

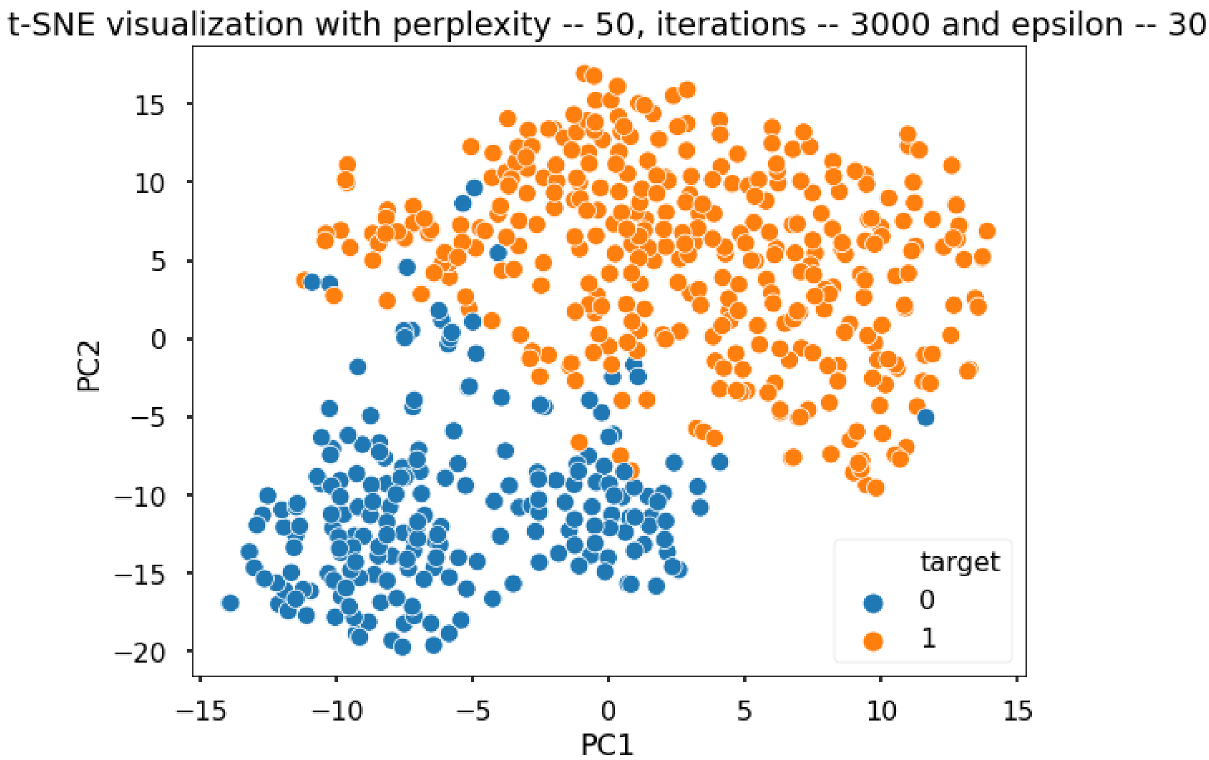

t-SNE visualization with perplexity -- 50, iterations -- 250 and epsilon -- 30
target
0
1
PC2
PC1
0.025
0.020
0.015
0.010
0.005
0.000
−0.005
−0.010
−0.015
−0.2
0.0
0.2
0.4
0.6

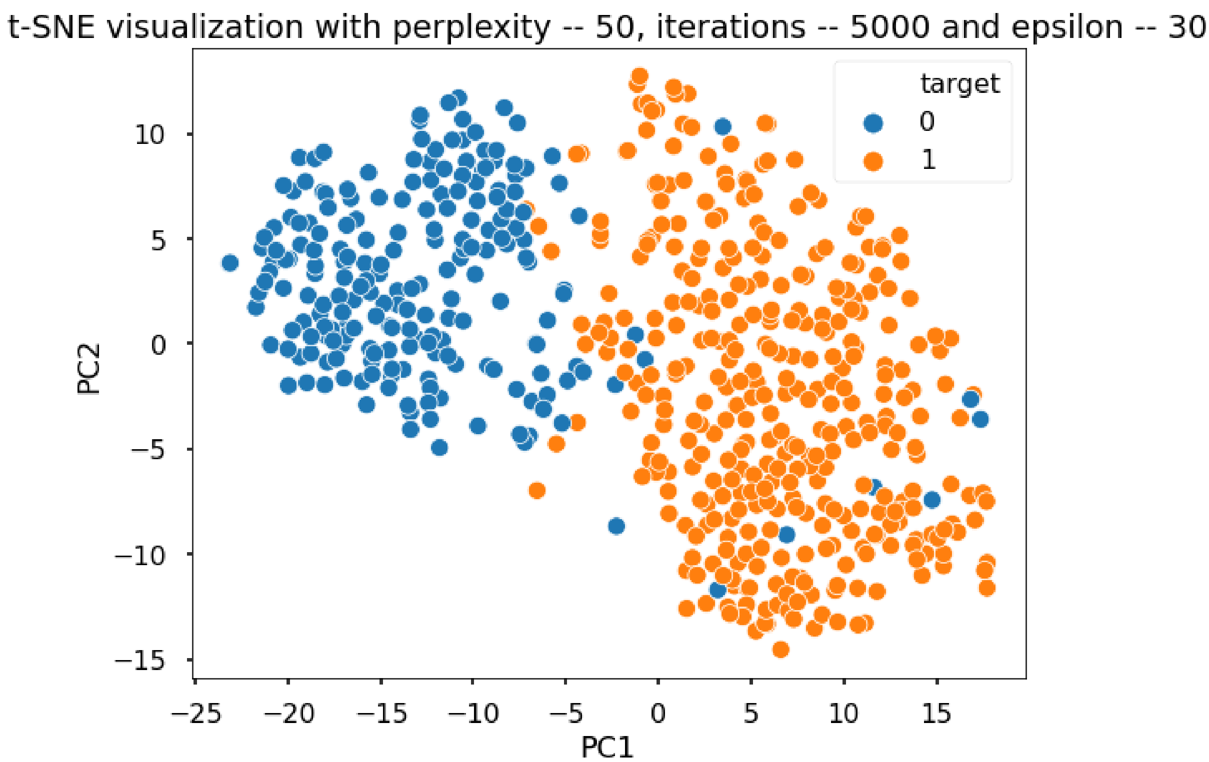

t-SNE visualization with perplexity -- 50, iterations -- 500 and epsilon -- 30
target
0
1
PC2
PC1
15
10
5
0
−5
−10
−15
−20
−10
−5
0
5
10

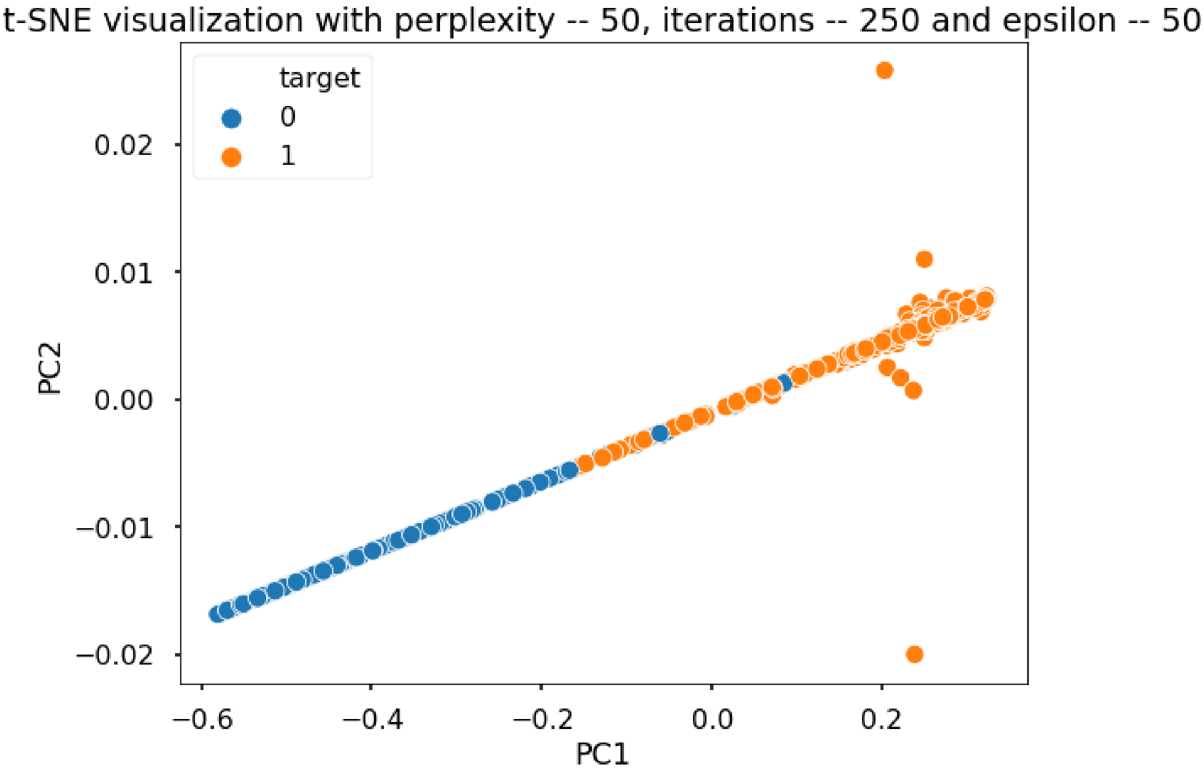

t-SNE visualization with perplexity -- 50, iterations -- 750 and epsilon -- 30
target
0
1
PC2
PC1
15
10
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
15

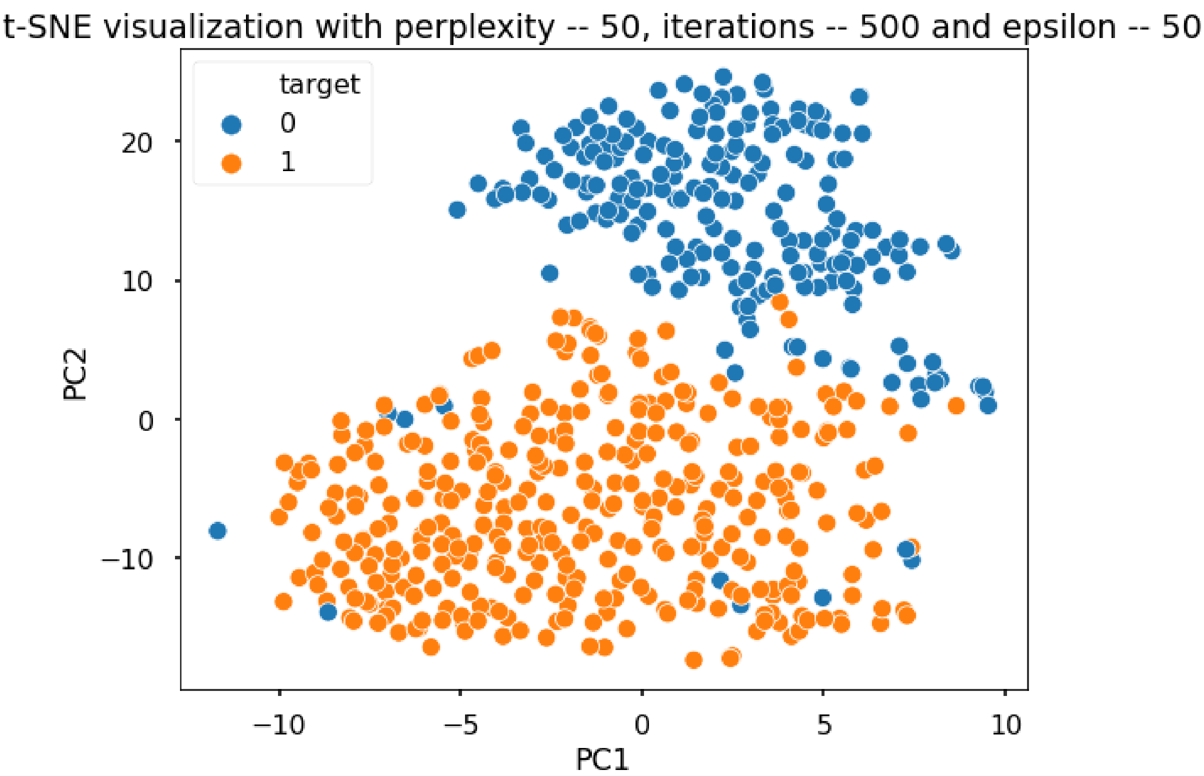

t-SNE visualization with perplexity -- 50, iterations -- 1000 and epsilon -- 30
target
0
1
PC2
PC1
20
15
10
5
0
−5
−10
−15
−10
−5
0
5
10


t-SNE visualization with perplexity -- 50, iterations -- 2000 and epsilon -- 30
target
0
1
PC2
PC1
15
10
5
0
−5
−10
−15
−20
−15
−10
−5
0
5
10
15

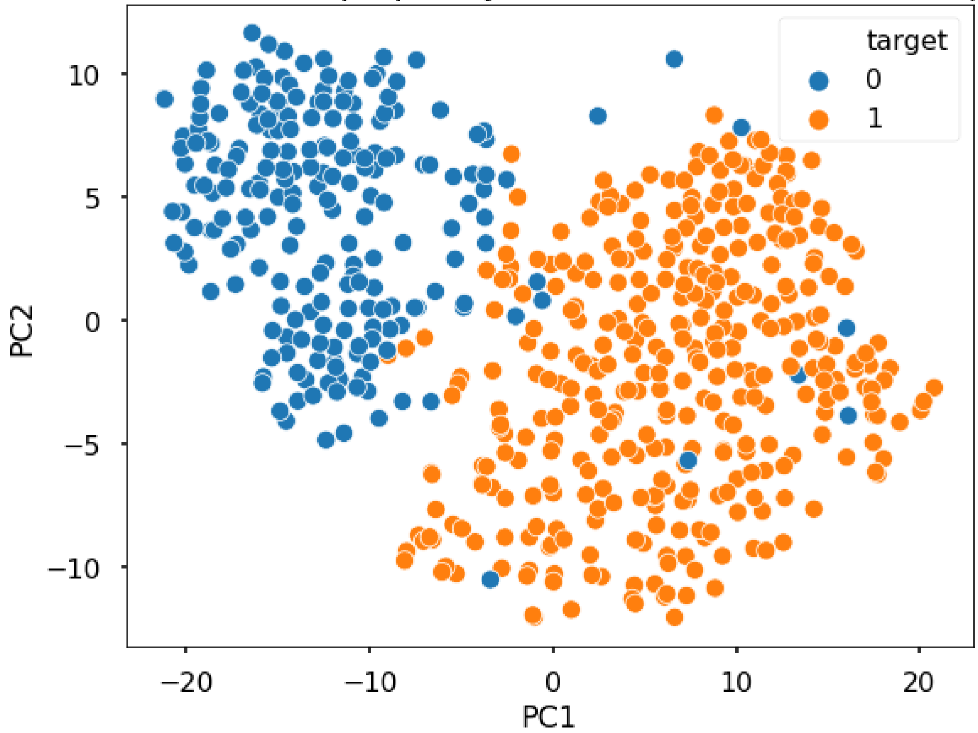

t-SNE visualization with perplexity -- 50, iterations -- 3000 and epsilon -- 30
PC2
PC1
target
0
1


t-SNE visualization with perplexity -- 50, iterations -- 5000 and epsilon -- 30
PC2
PC1
target
0
1

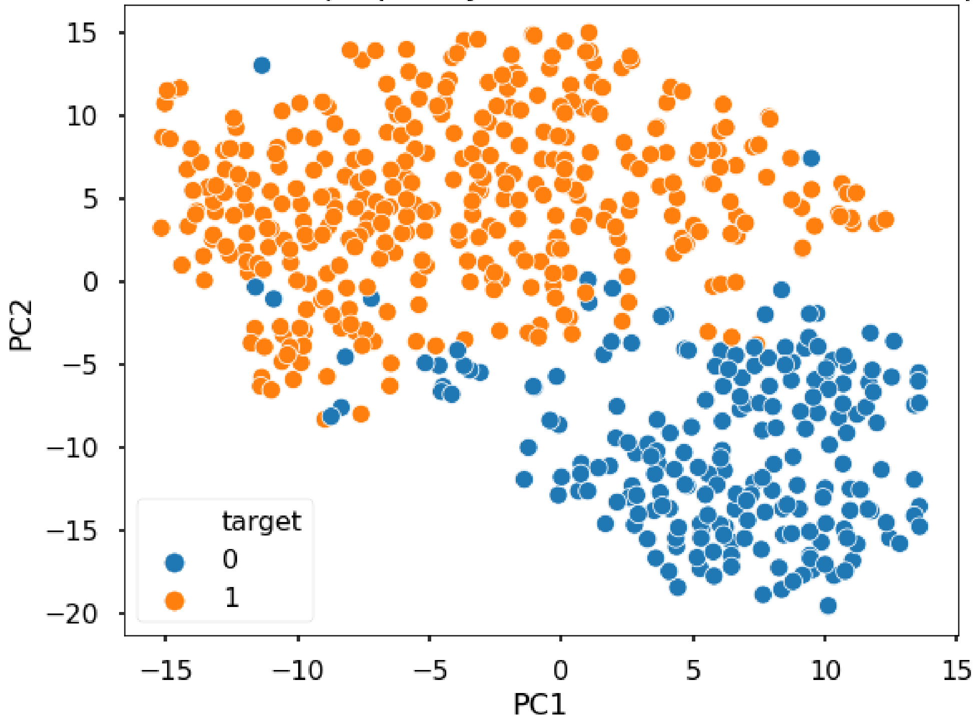

t-SNE visualization with perplexity -- 50, iterations -- 250 and epsilon -- 50
target
0
1
0.02
0.01
0.00
−0.01
−0.02
PC2
−0.6
−0.4
−0.2
0.0
0.2
PC1

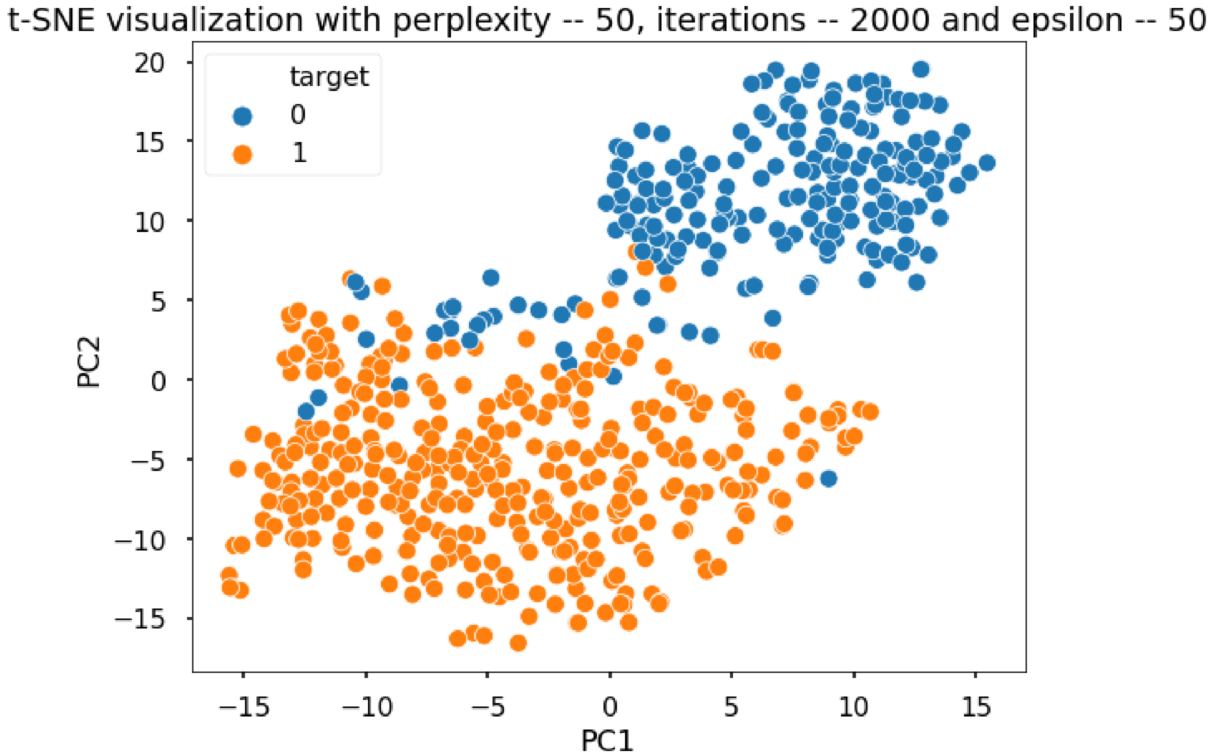

t-SNE visualization with perplexity -- 50, iterations -- 500 and epsilon -- 50
target
0
1
20
10
0
−10
PC2
−10
−5
0
5
10
PC1

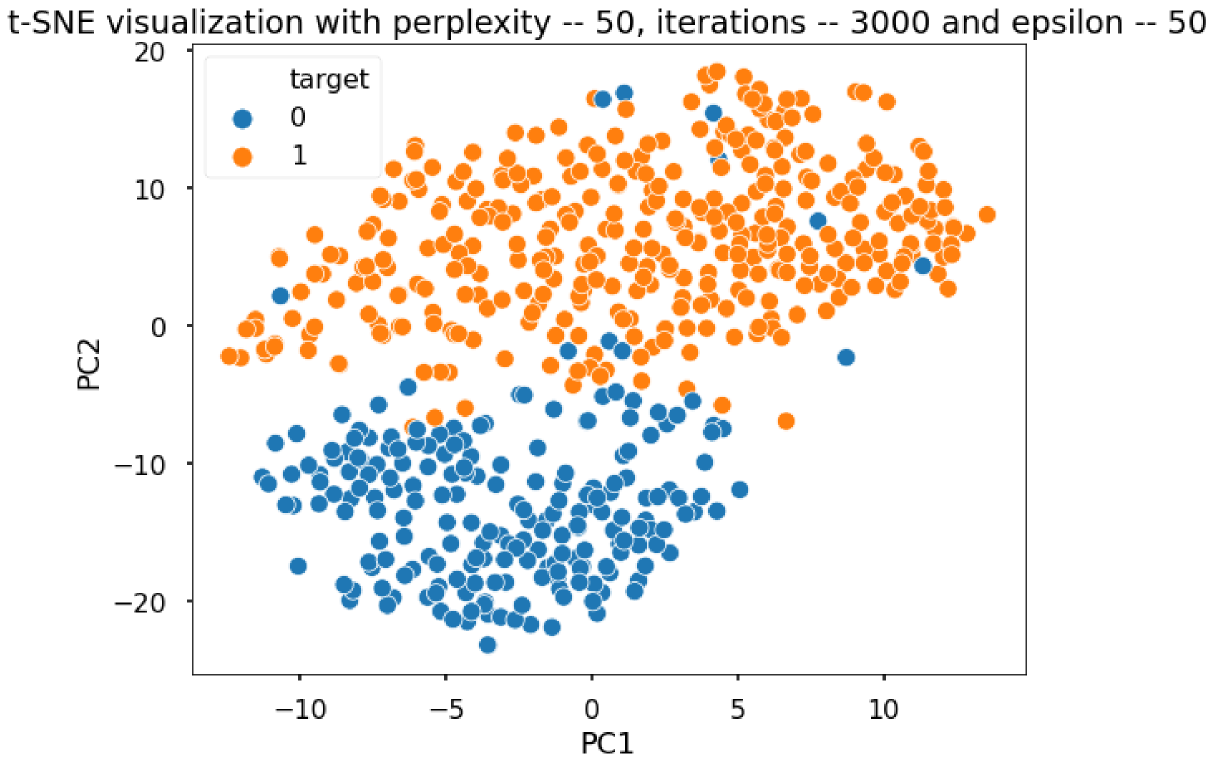
t-SNE visualization with perplexity -- 50, iterations -- 750 and epsilon -- 50

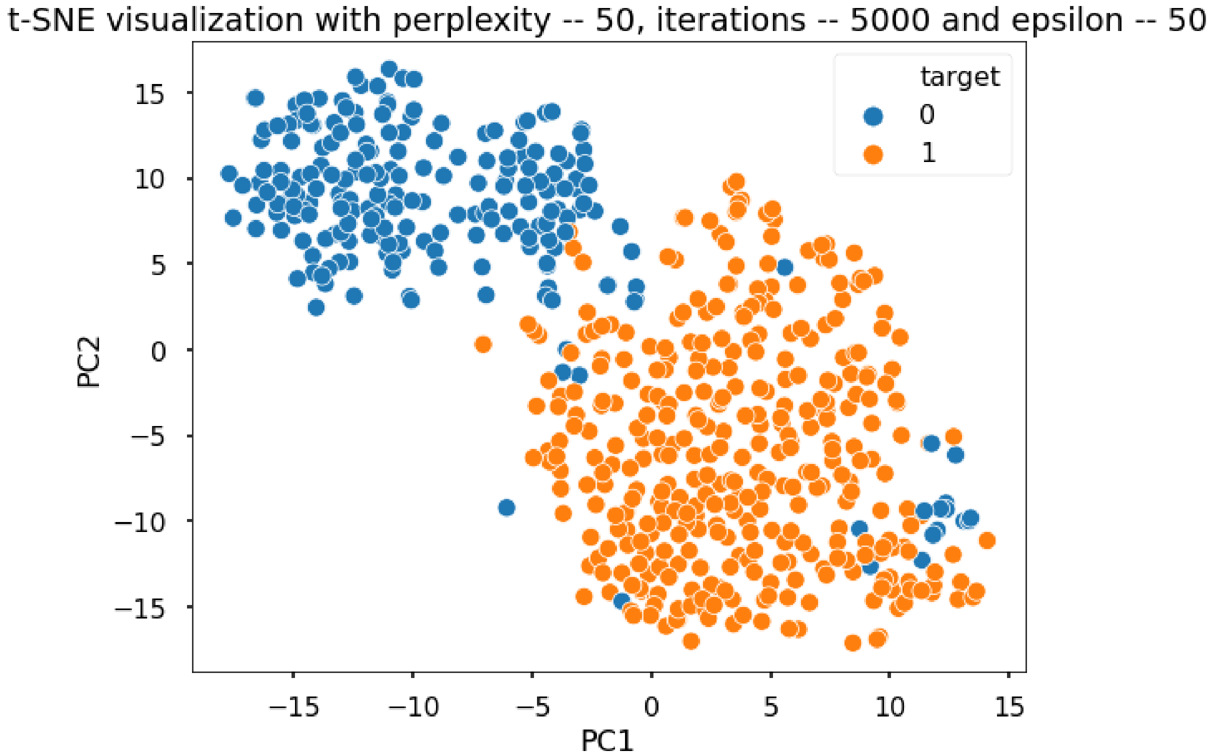


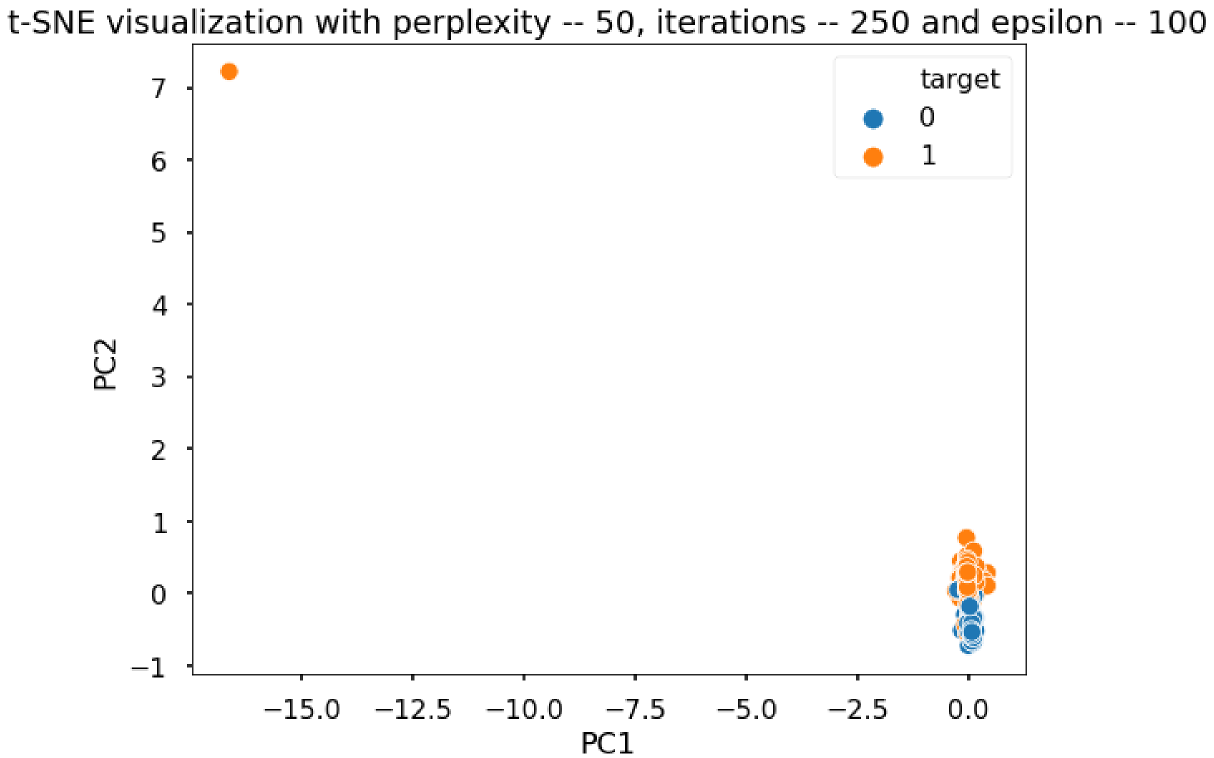
t-SNE visualization with perplexity -- 50, iterations -- 1000 and epsilon -- 50



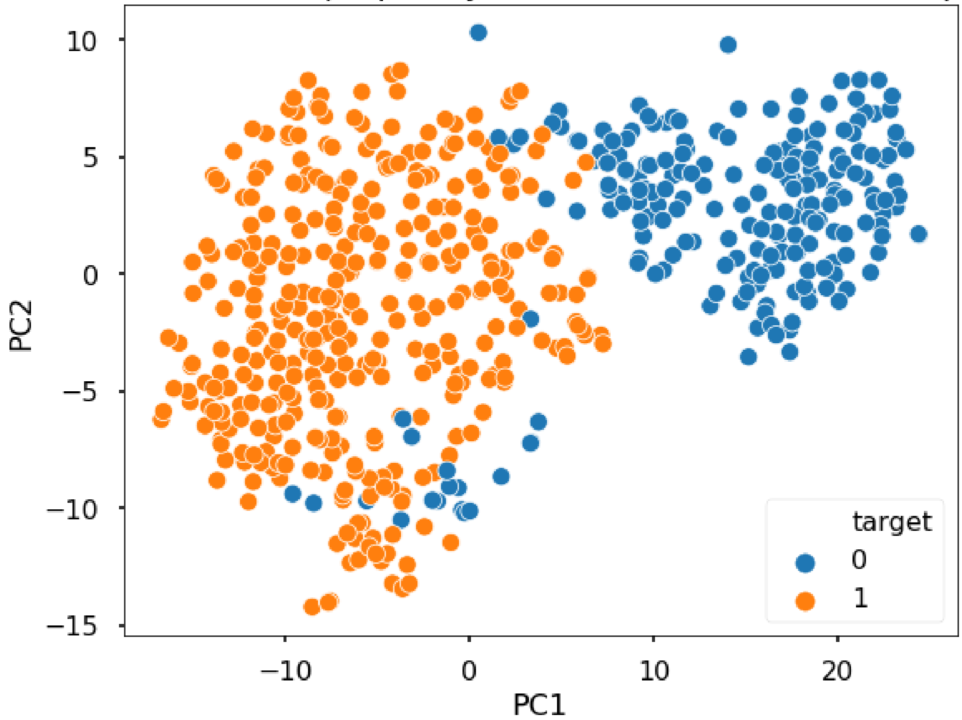

t-SNE visualization with perplexity -- 50, iterations -- 2000 and epsilon -- 50
target
0
1
PC1
PC2


t-SNE visualization with perplexity -- 50, iterations -- 3000 and epsilon -- 50
target
0
1
PC1
PC2

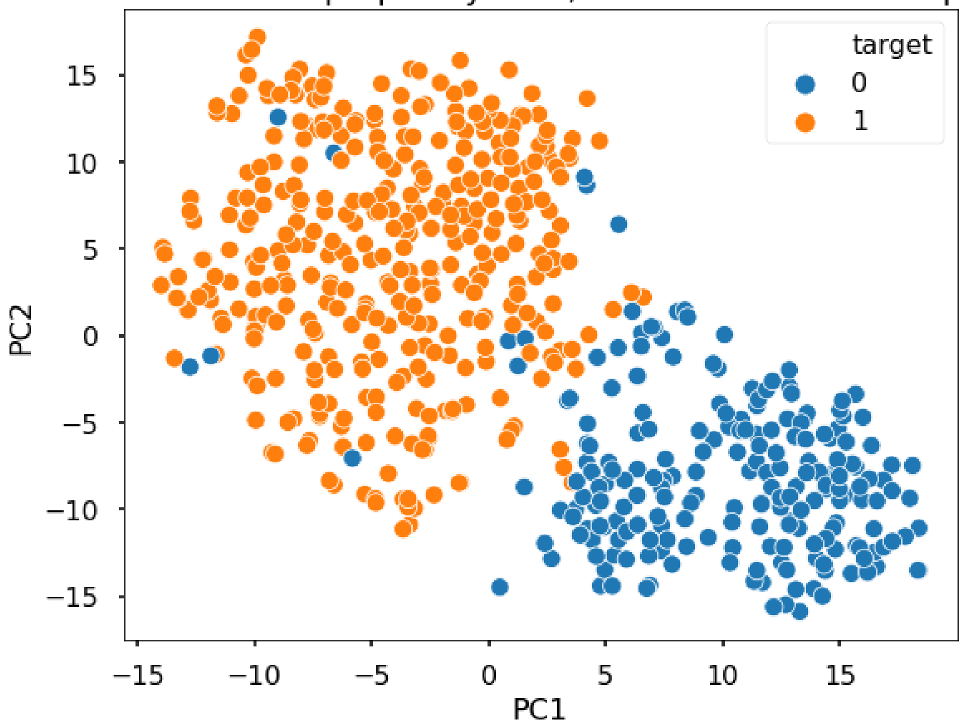

t-SNE visualization with perplexity -- 50, iterations -- 5000 and epsilon -- 50
target
0
1
PC2
PC1
15
10
5
0
−5
−10
−15
−15
−10
−5
0
5
10
15

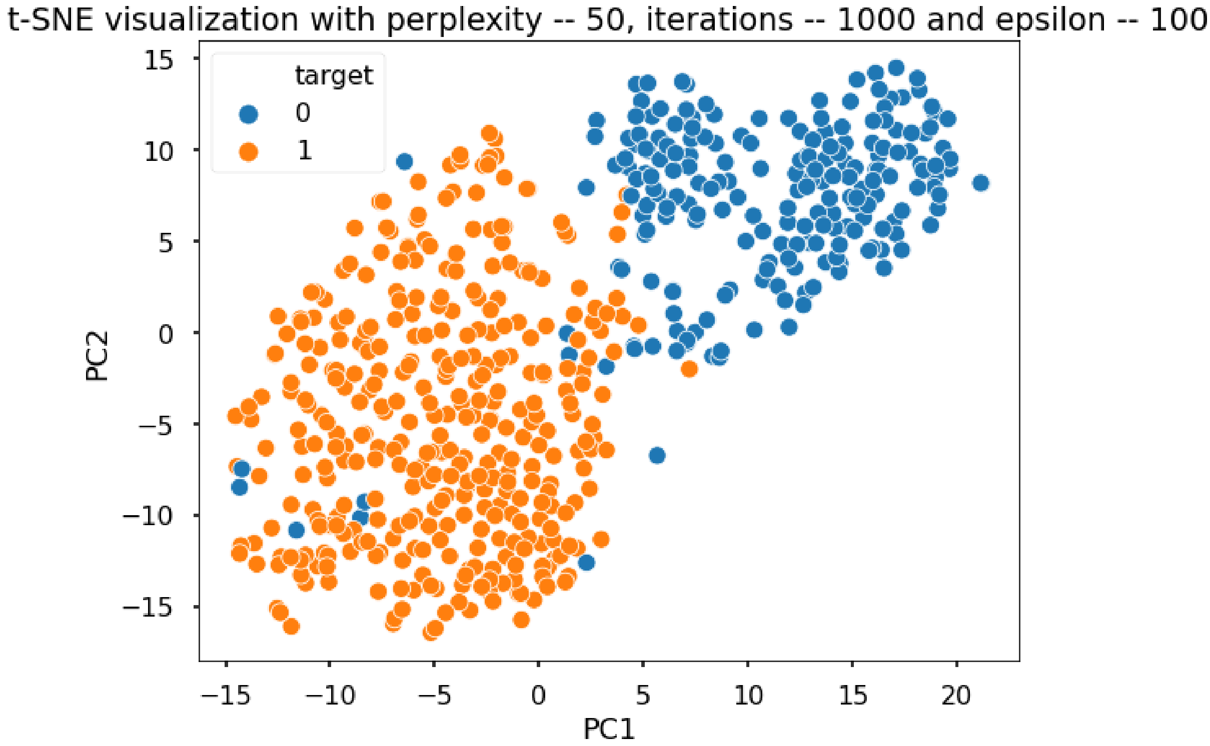

t-SNE visualization with perplexity -- 50, iterations -- 250 and epsilon -- 100
target
0
1
PC2
PC1
7
6
5
4
3
2
1
0
−1
−15.0
−12.5
−10.0
−7.5
−5.0
−2.5
0.0

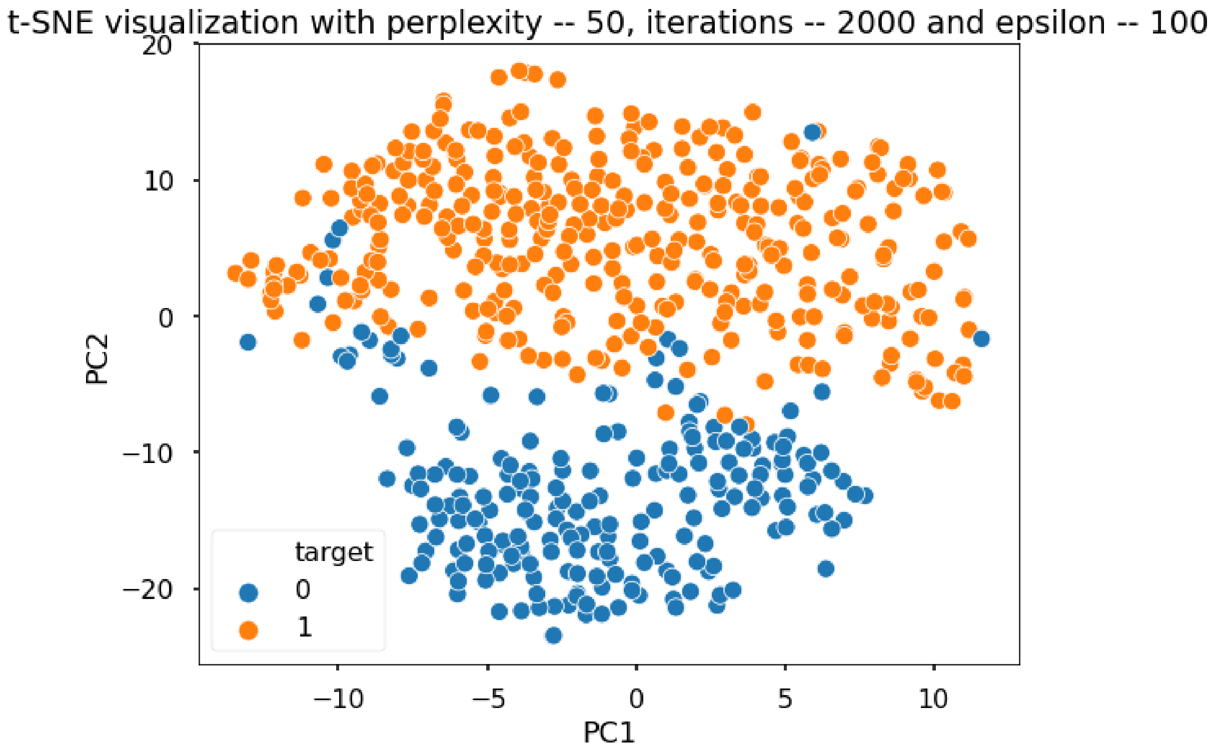
t-SNE visualization with perplexity -- 50, iterations -- 500 and epsilon -- 100



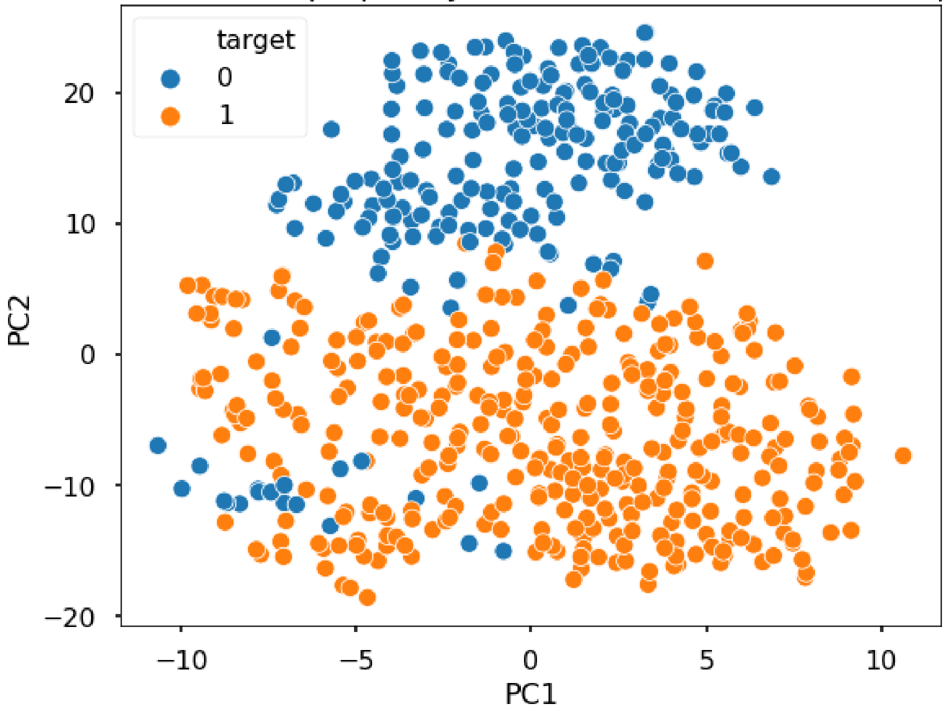
t-SNE visualization with perplexity -- 50, iterations -- 750 and epsilon -- 100



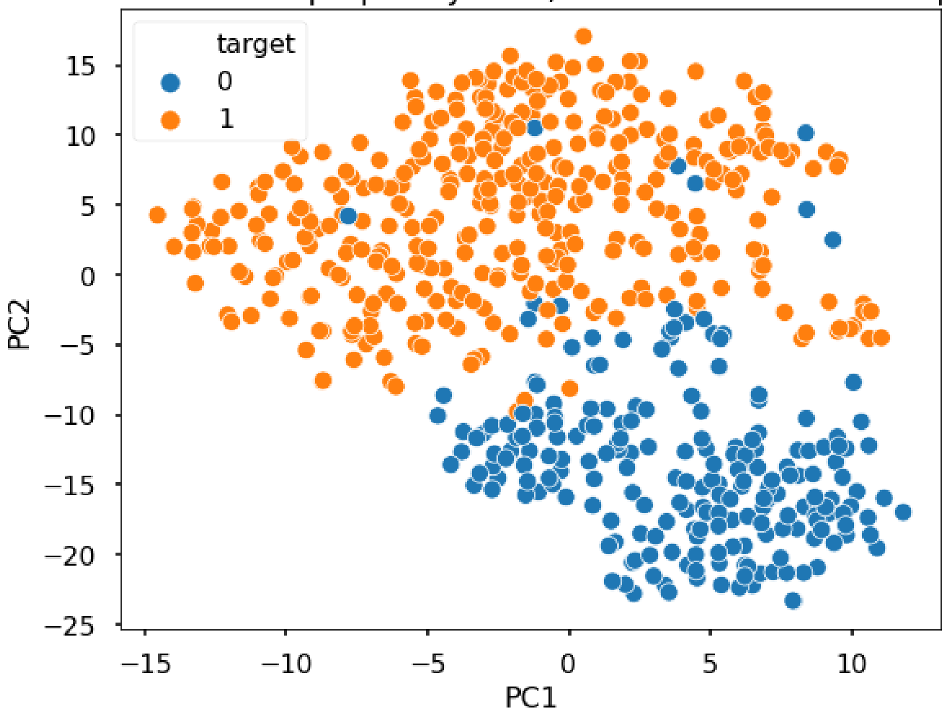

t-SNE visualization with perplexity -- 50, iterations -- 1000 and epsilon -- 100
target
0
1
PC2
PC1

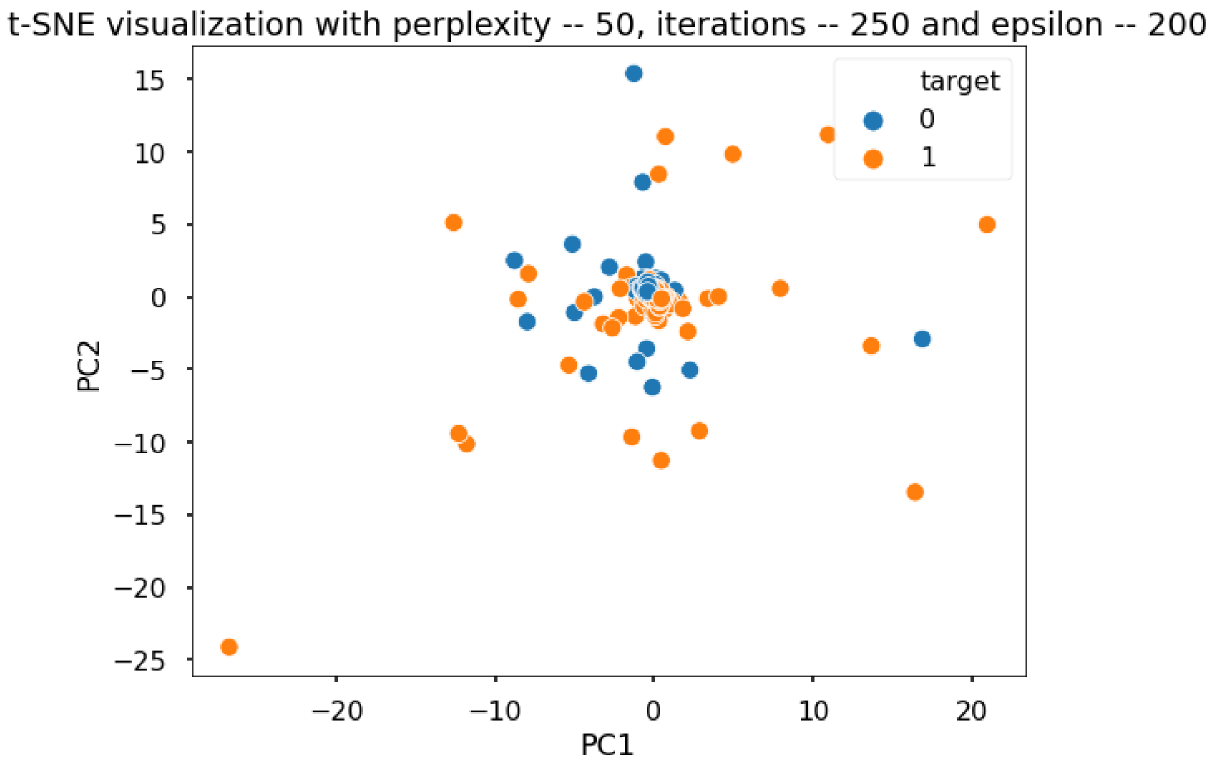

t-SNE visualization with perplexity -- 50, iterations -- 2000 and epsilon -- 100
target
0
1
PC2
PC1

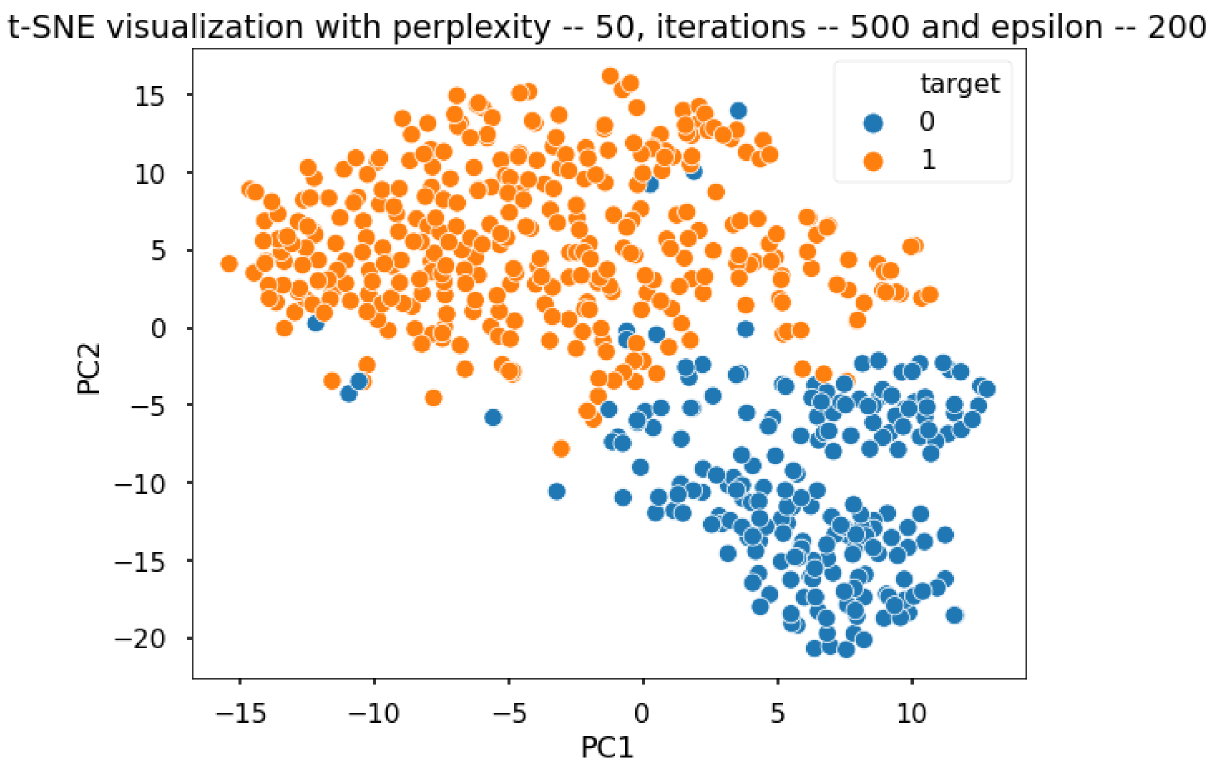
t-SNE visualization with perplexity -- 50, iterations -- 3000 and epsilon -- 100



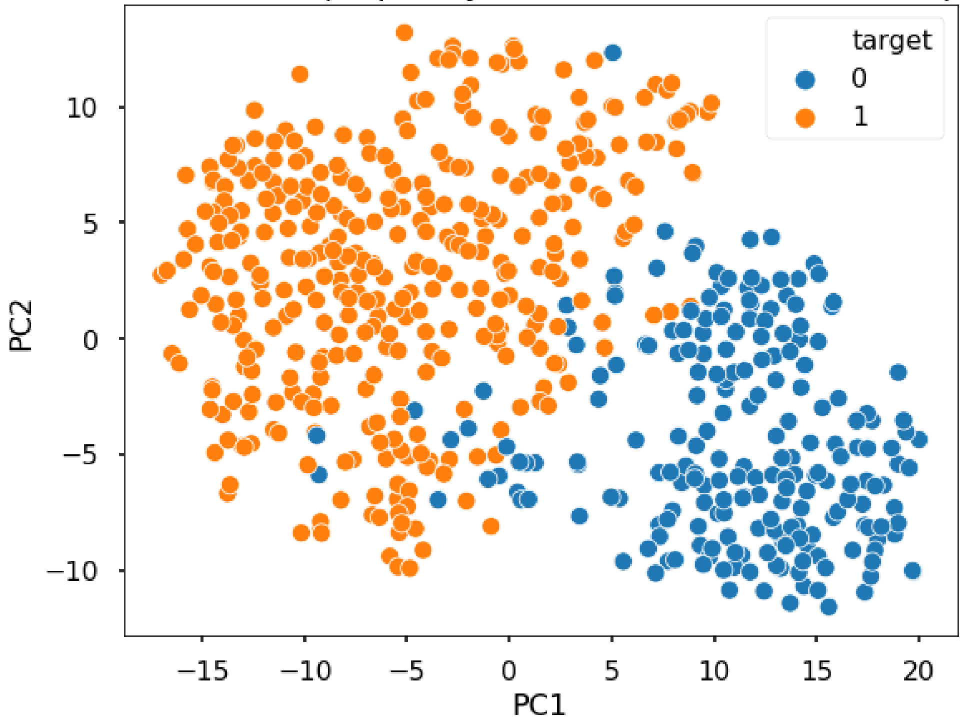
t-SNE visualization with perplexity -- 50, iterations -- 5000 and epsilon -- 100



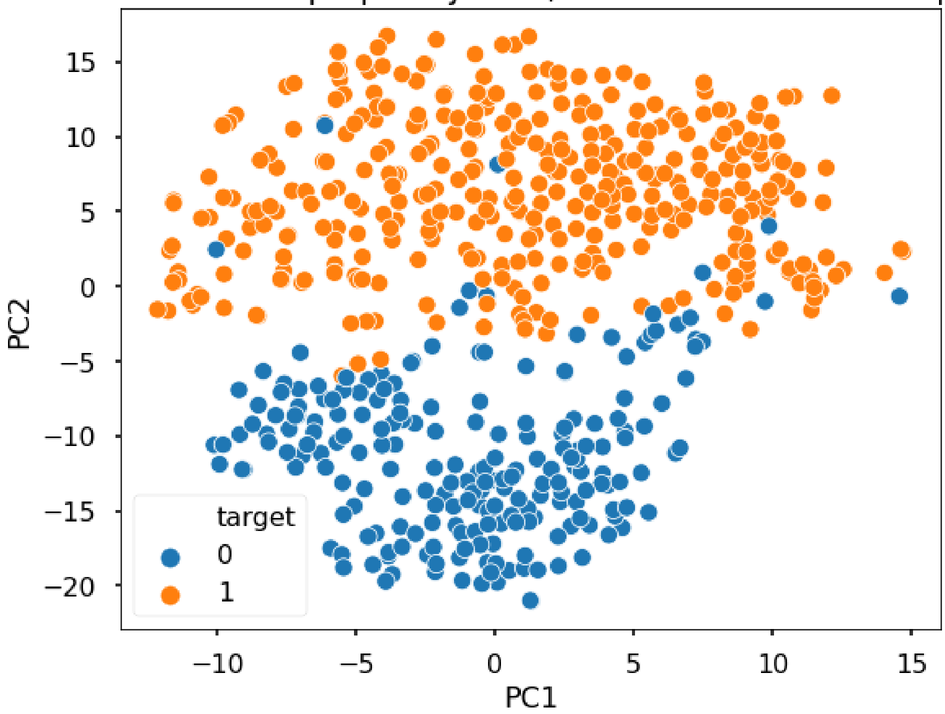

t-SNE visualization with perplexity -- 50, iterations -- 250 and epsilon -- 200
target
0
1
PC2
PC1


t-SNE visualization with perplexity -- 50, iterations -- 500 and epsilon -- 200
target
0
1
PC2
PC1

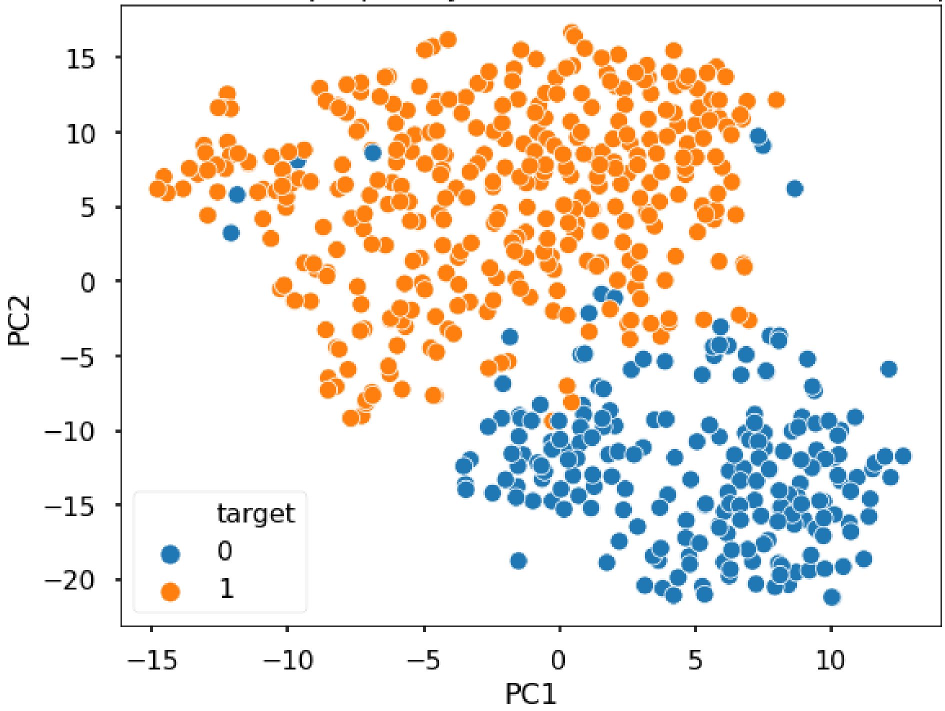
t-SNE visualization with perplexity -- 50, iterations -- 750 and epsilon -- 200



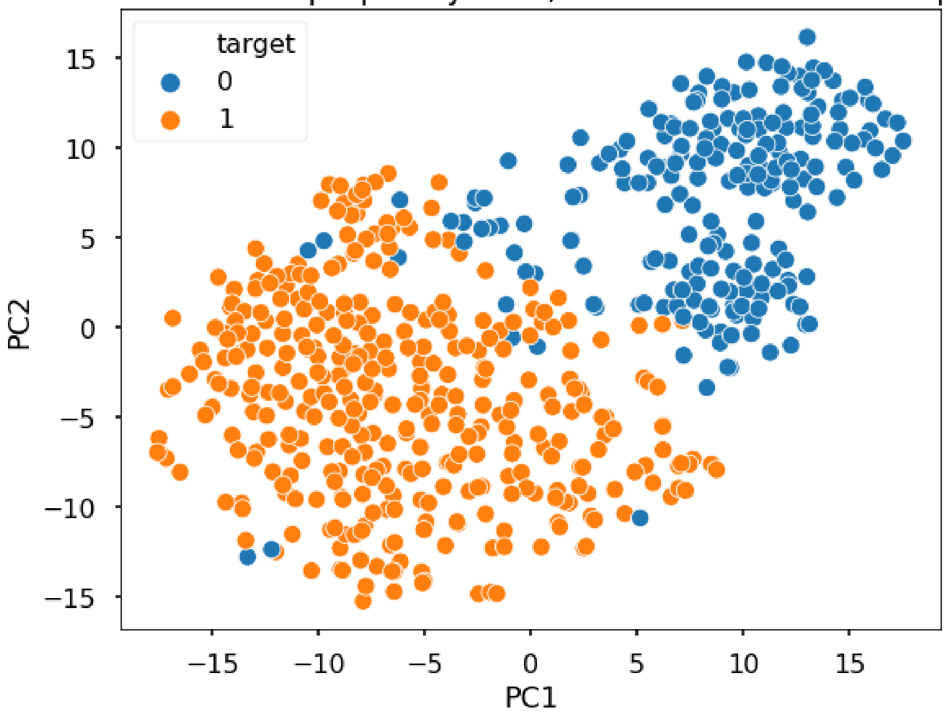
t-SNE visualization with perplexity -- 50, iterations -- 1000 and epsilon -- 200



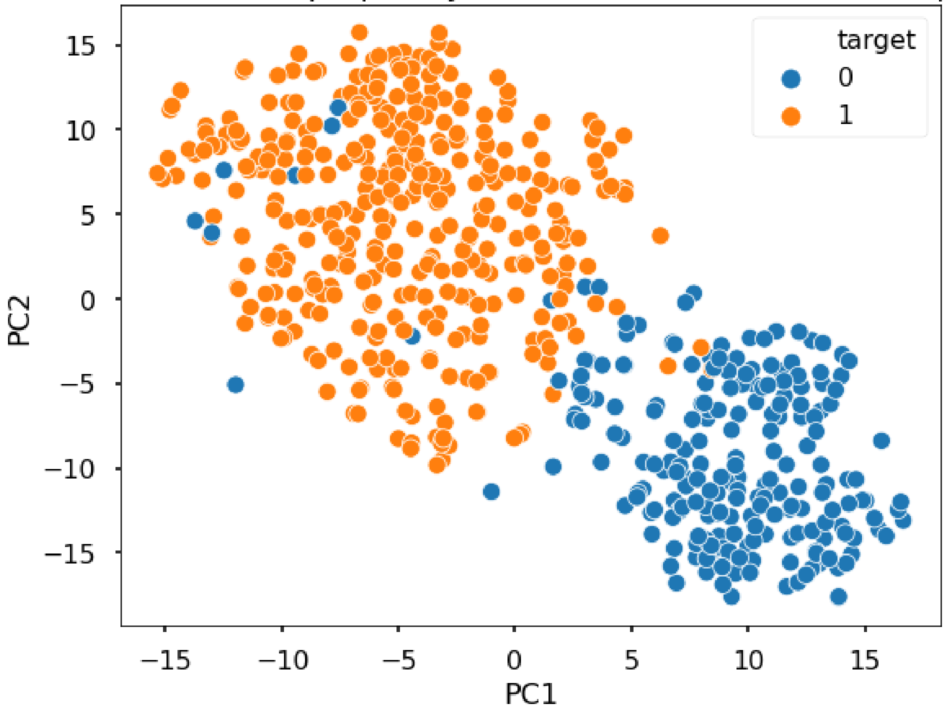
t-SNE visualization with perplexity -- 50, iterations -- 2000 and epsilon -- 200



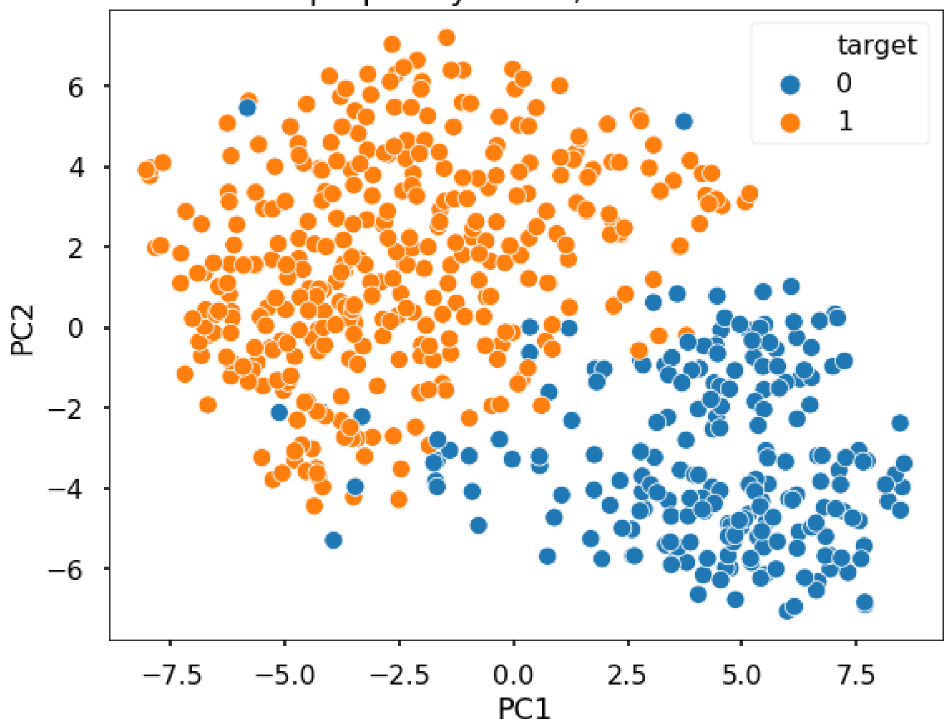
t-SNE visualization with perplexity -- 50, iterations -- 3000 and epsilon -- 200

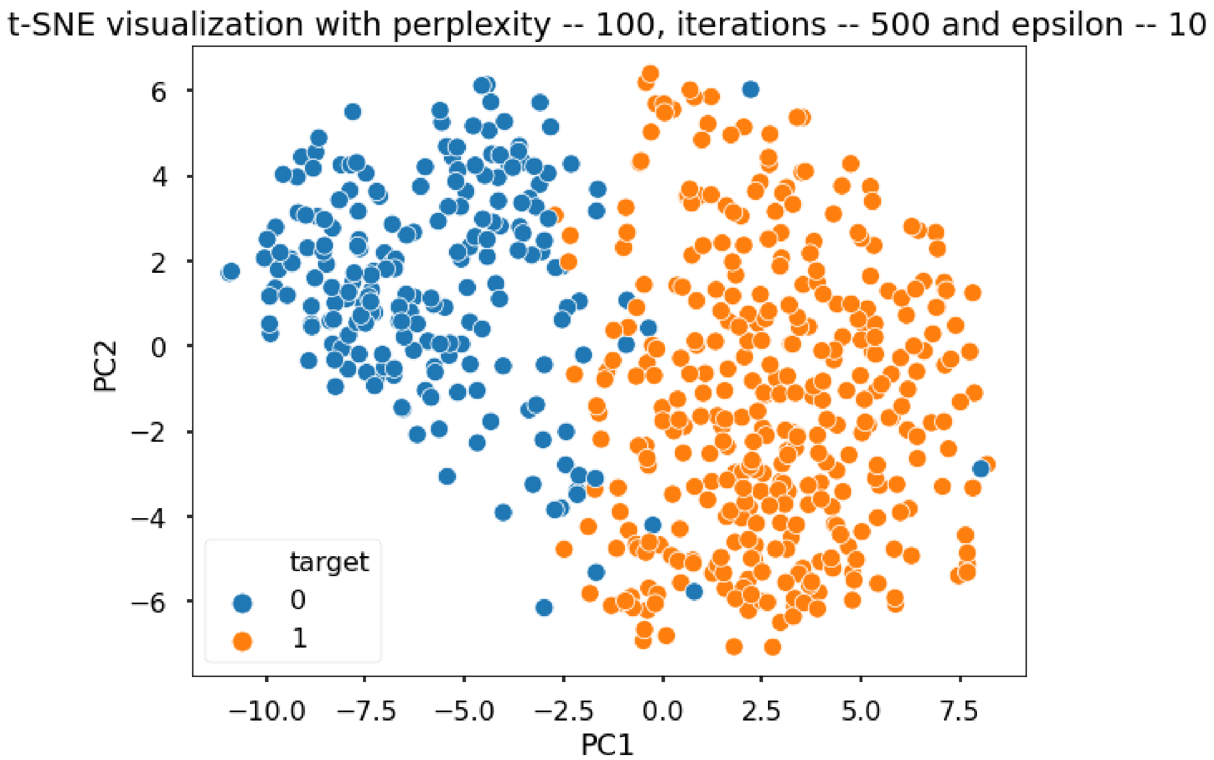


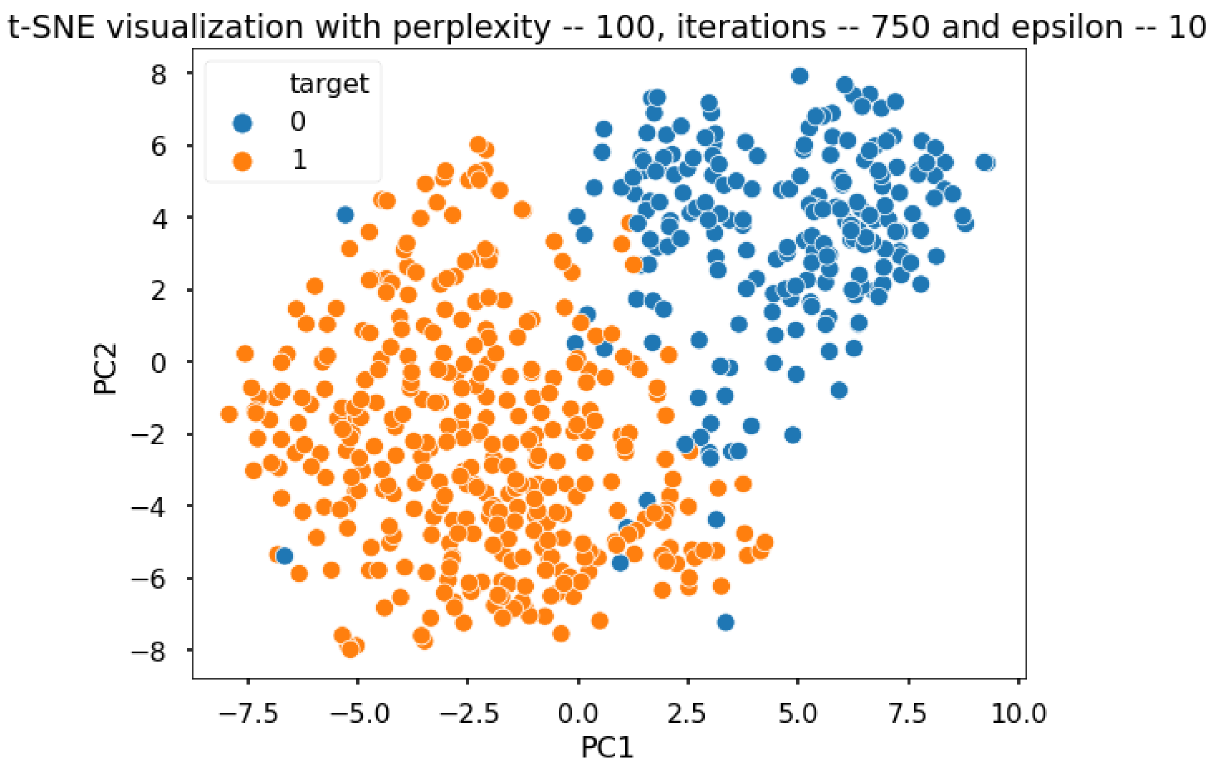
t-SNE visualization with perplexity -- 50, iterations -- 5000 and epsilon -- 200

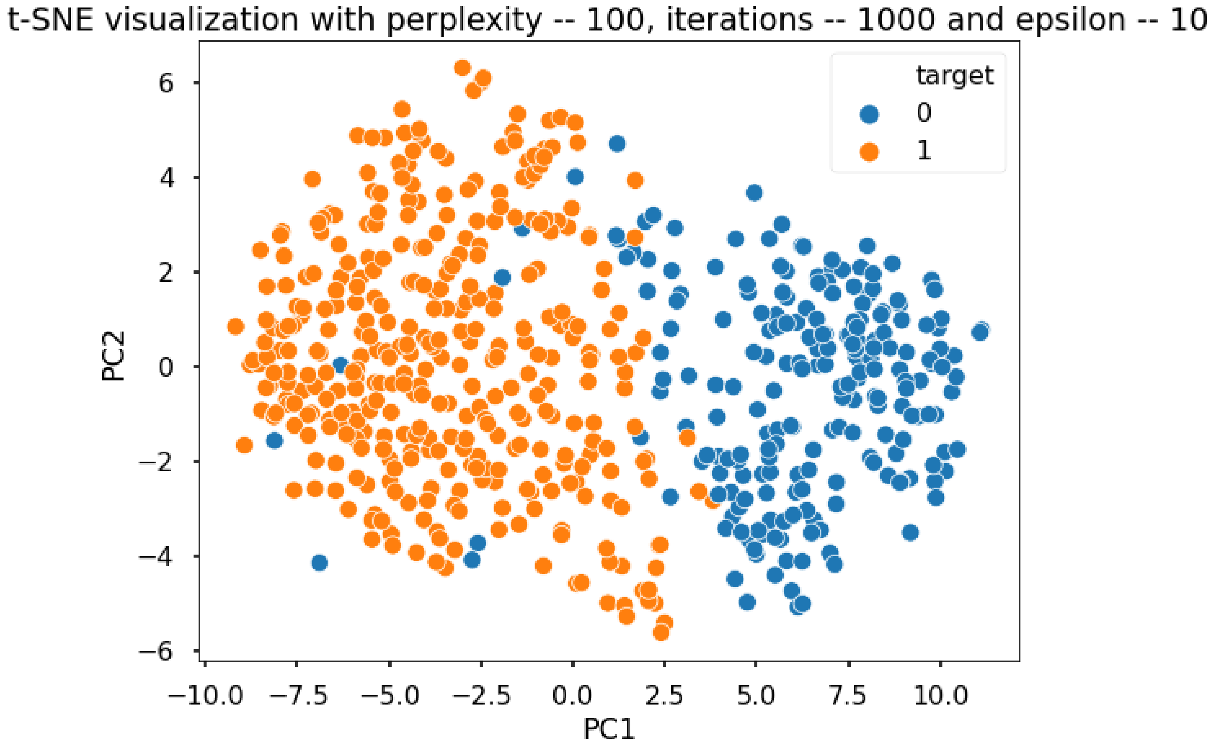


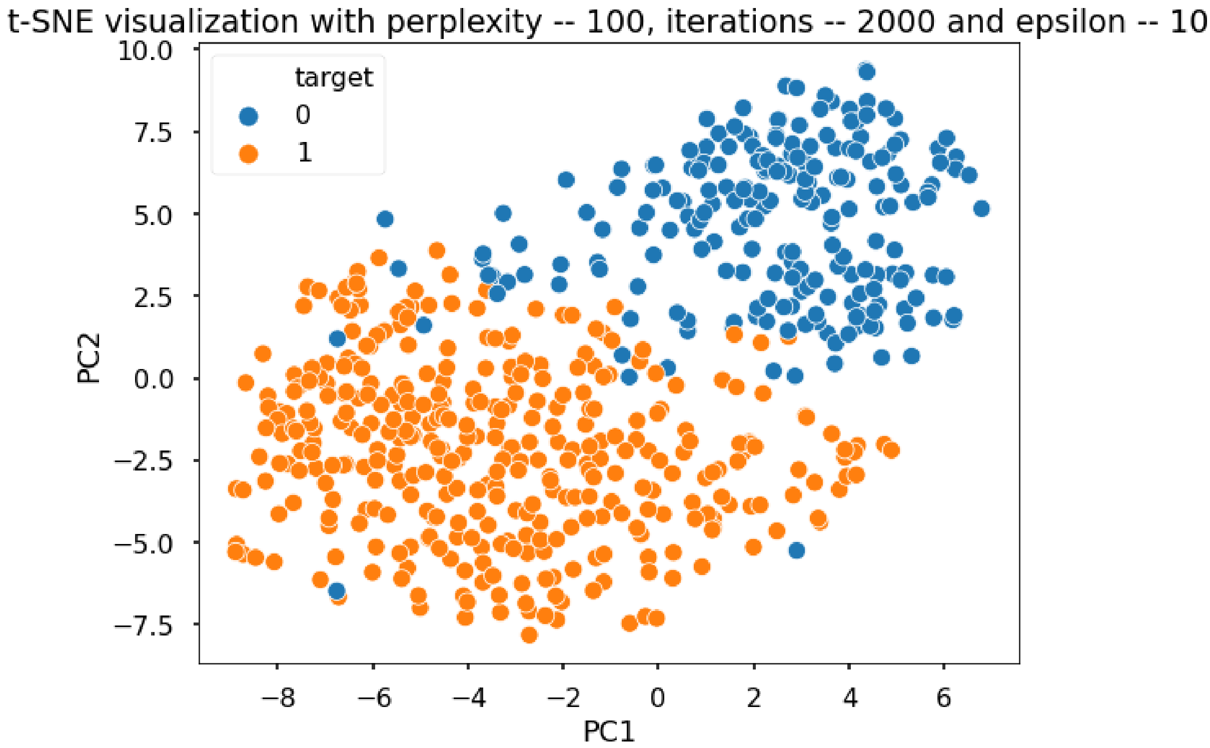
t-SNE visualization with perplexity -- 100, iterations -- 250 and epsilon -- 10

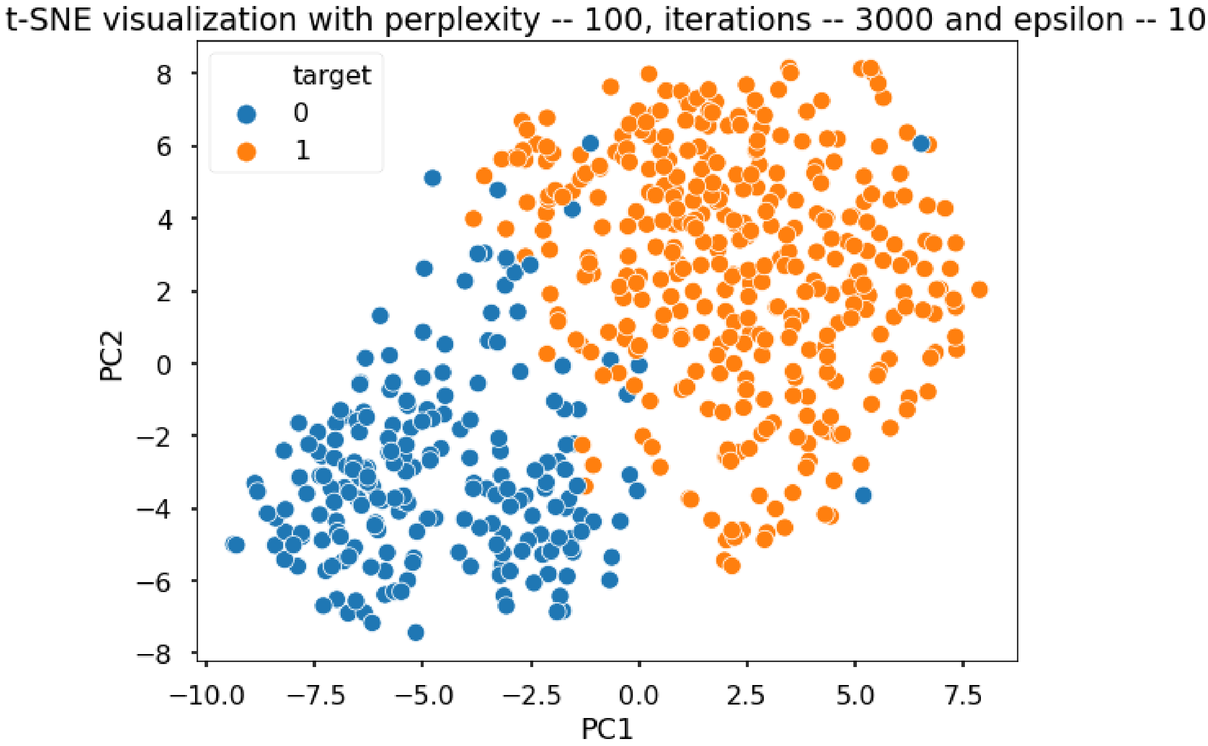


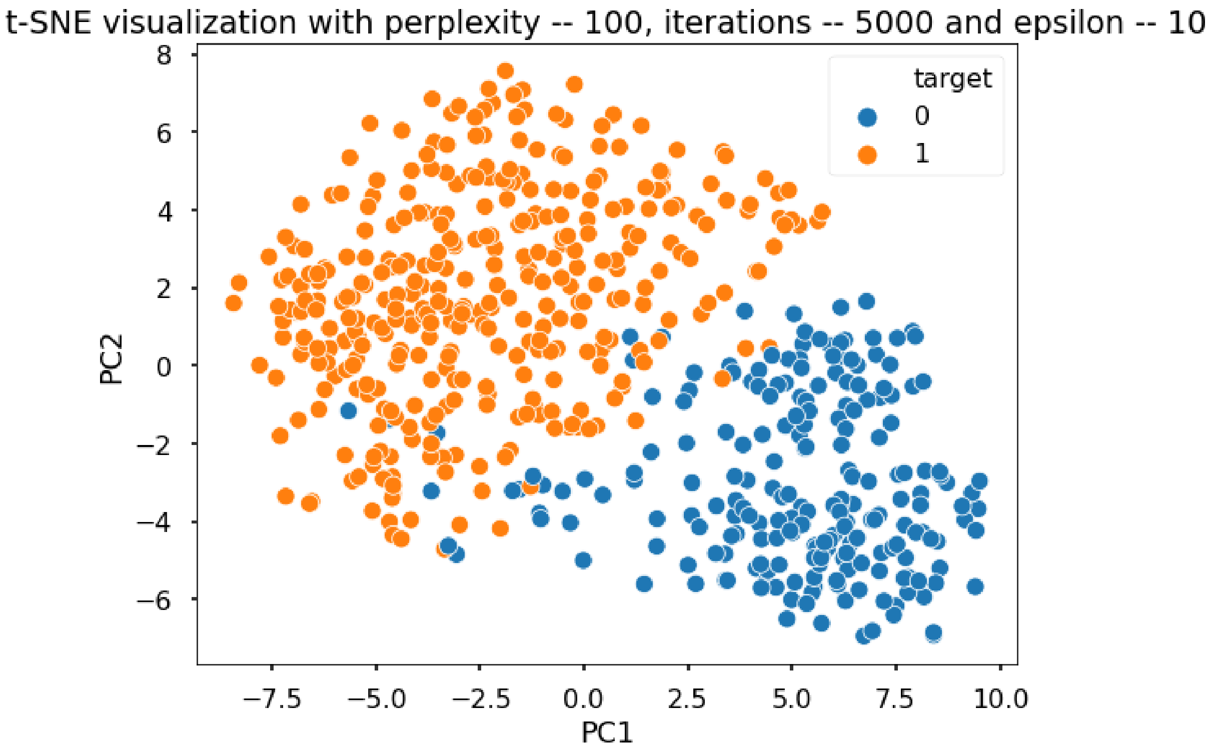

t-SNE visualization with perplexity -- 100, iterations -- 500 and epsilon -- 10
PC2
PC1
target
0
1
6
4
2
0
−2
−4
−6
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5

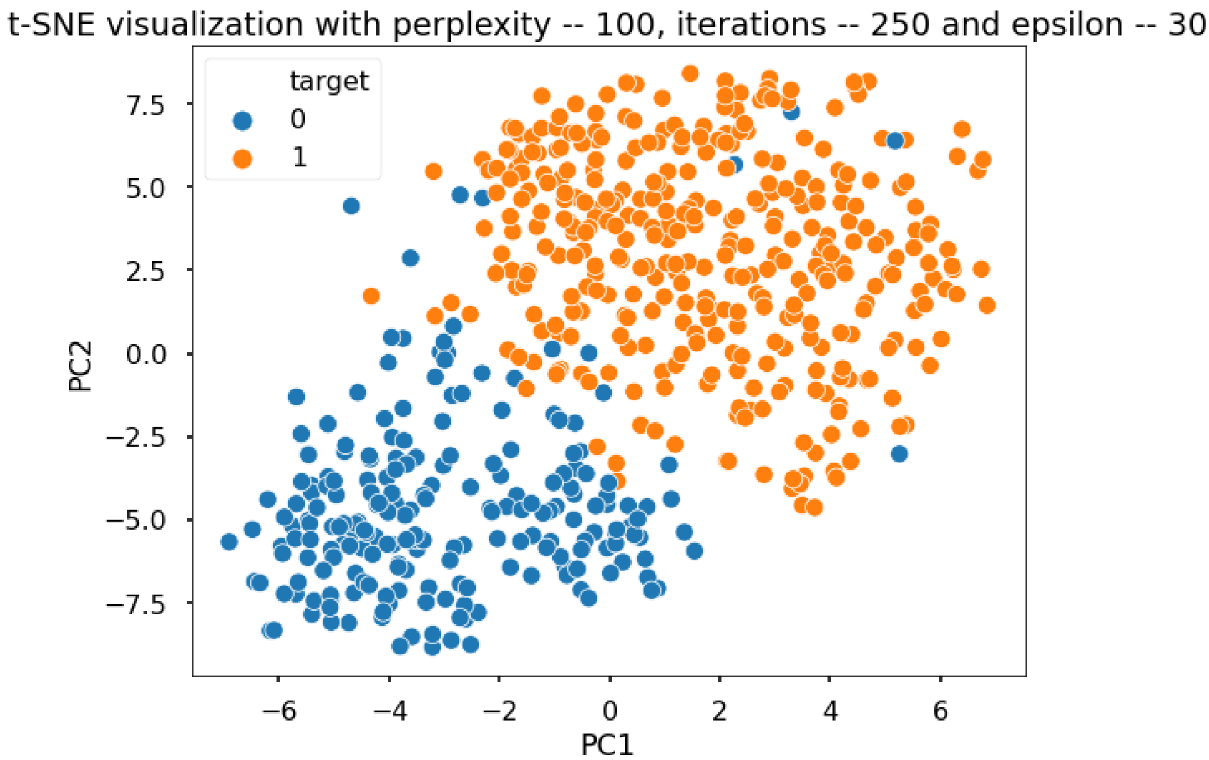

t-SNE visualization with perplexity -- 100, iterations -- 750 and epsilon -- 10
PC2
PC1
target
0
1
8
6
4
2
0
−2
−4
−6
−8
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

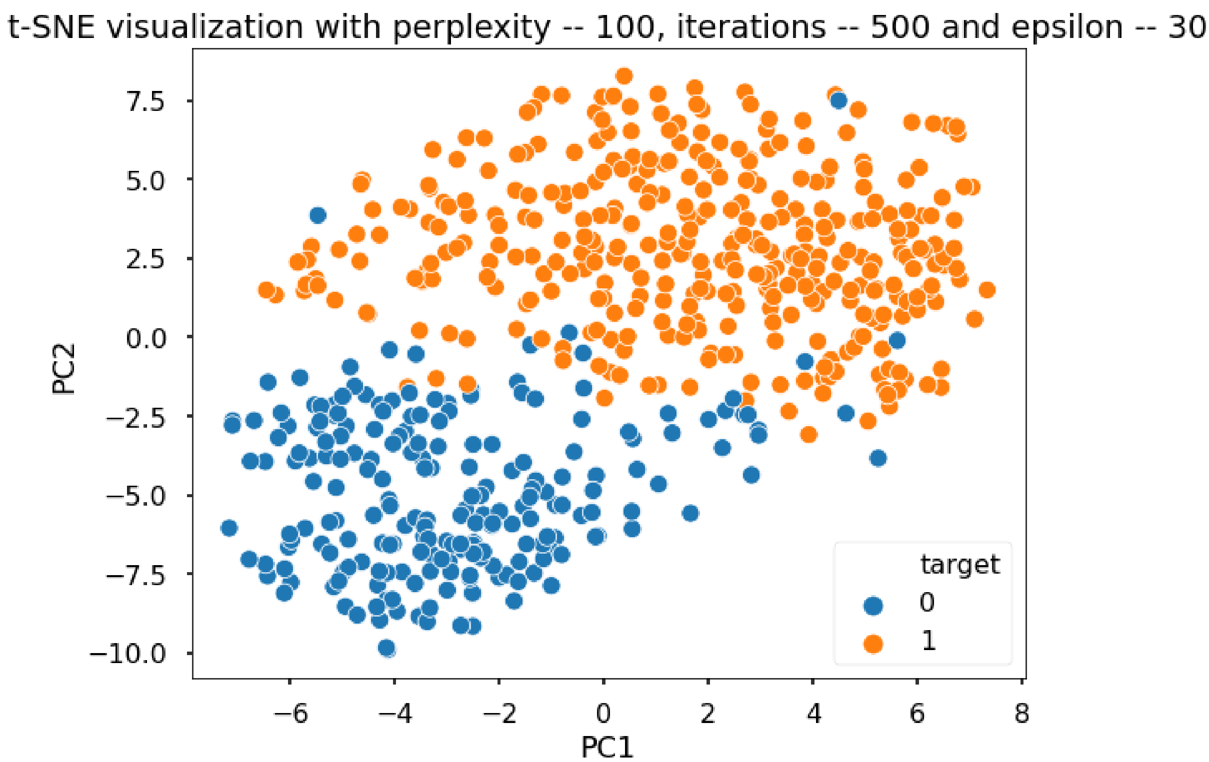

t-SNE visualization with perplexity -- 100, iterations -- 1000 and epsilon -- 10
target
0
1
PC2
PC1
6
4
2
0
−2
−4
−6
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

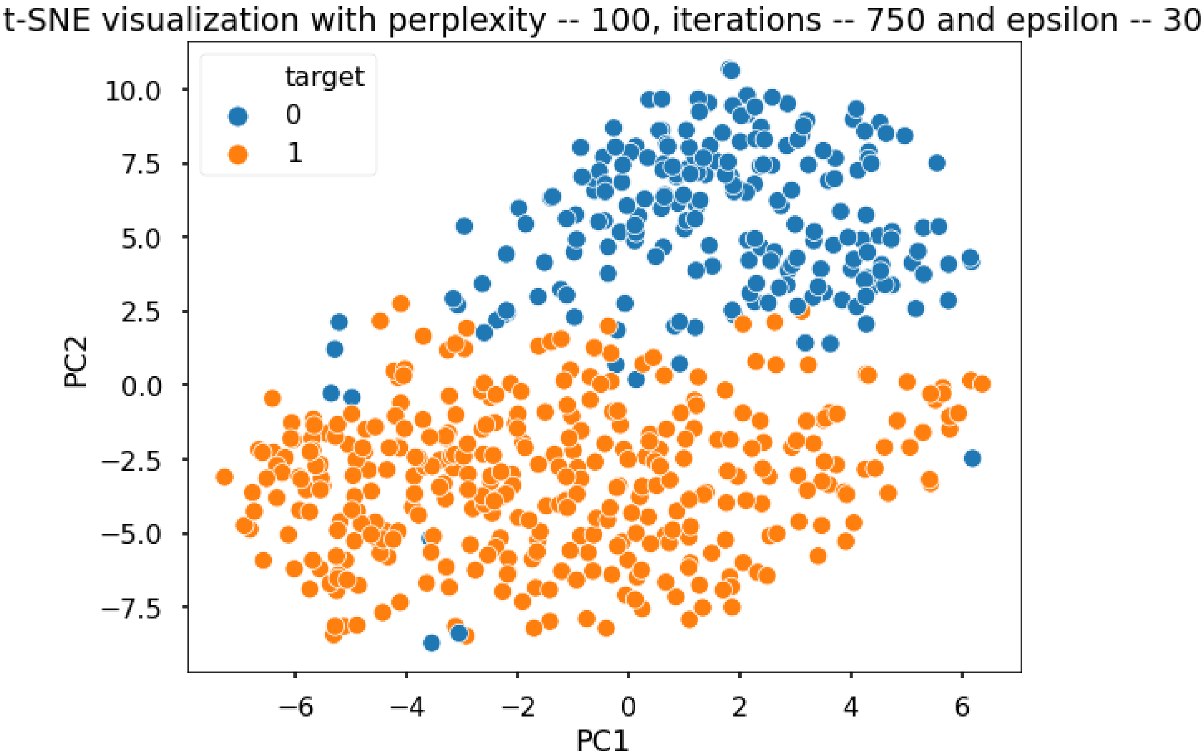

t-SNE visualization with perplexity -- 100, iterations -- 2000 and epsilon -- 10
target
0
1
PC2
PC1
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−8
−6
−4
−2
0
2
4
6

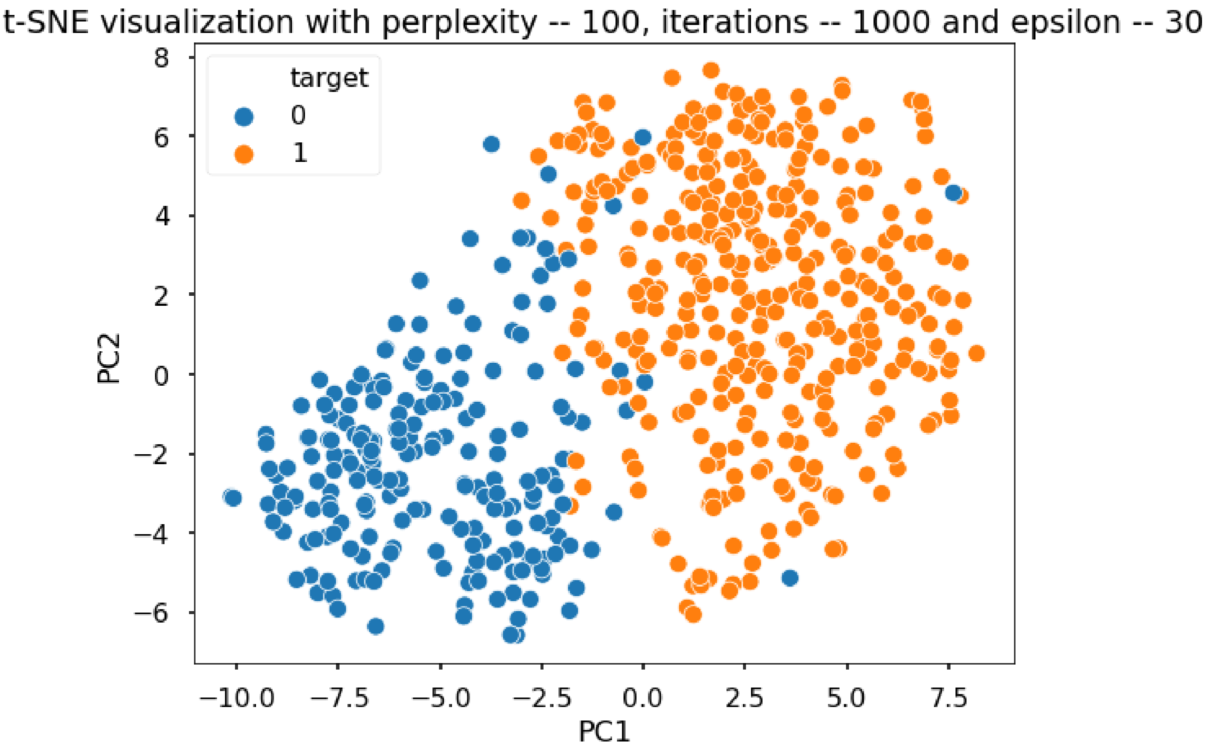

t-SNE visualization with perplexity -- 100, iterations -- 3000 and epsilon -- 10
target
0
1
PC2
PC1
8
6
4
2
0
−2
−4
−6
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5


t-SNE visualization with perplexity -- 100, iterations -- 5000 and epsilon -- 10
target
0
1
PC2
PC1
8
6
4
2
0
−2
−4
−6
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

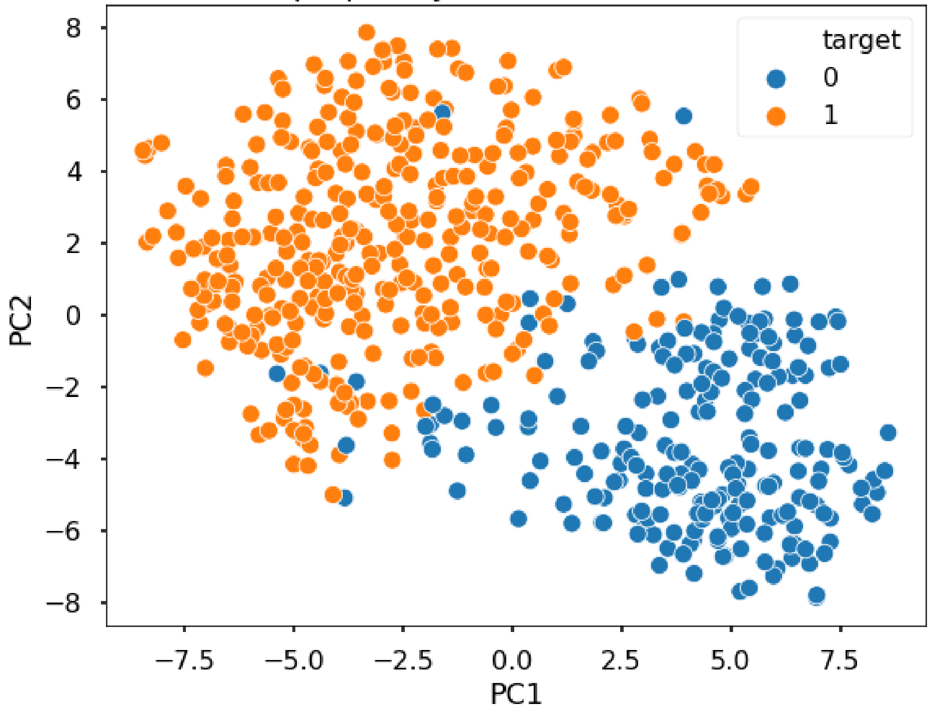

t-SNE visualization with perplexity -- 100, iterations -- 250 and epsilon -- 30
target
0
1
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
PC2
−6
−4
−2
0
2
4
6
PC1


t-SNE visualization with perplexity -- 100, iterations -- 500 and epsilon -- 30
target
0
1
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
PC2
−6
−4
−2
0
2
4
6
8
PC1

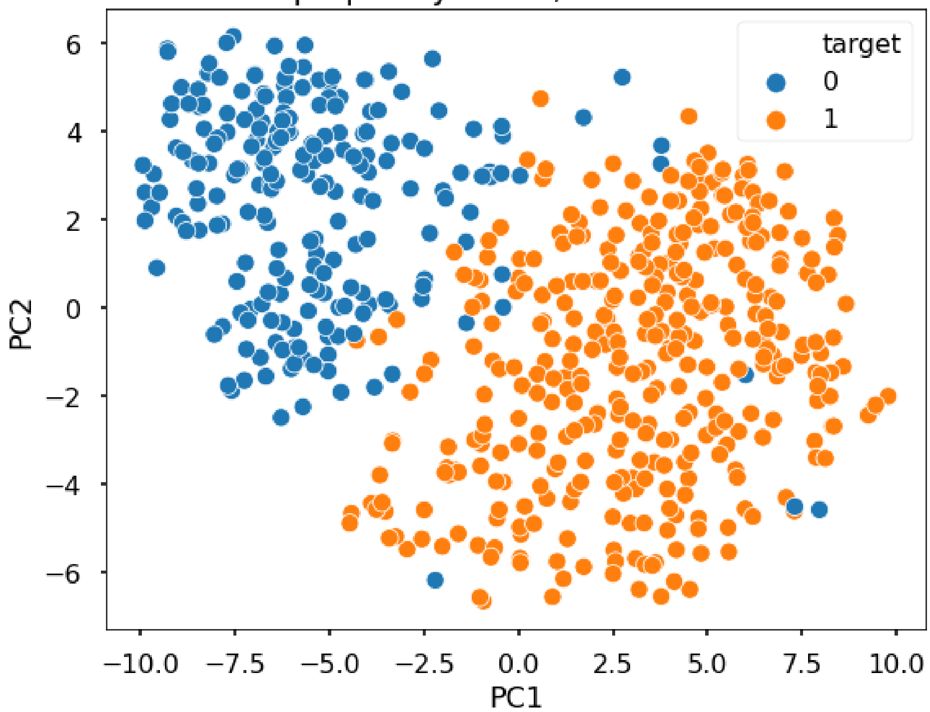

t-SNE visualization with perplexity -- 100, iterations -- 750 and epsilon -- 30
target
0
1
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
PC2
−6
−4
−2
0
2
4
6
PC1

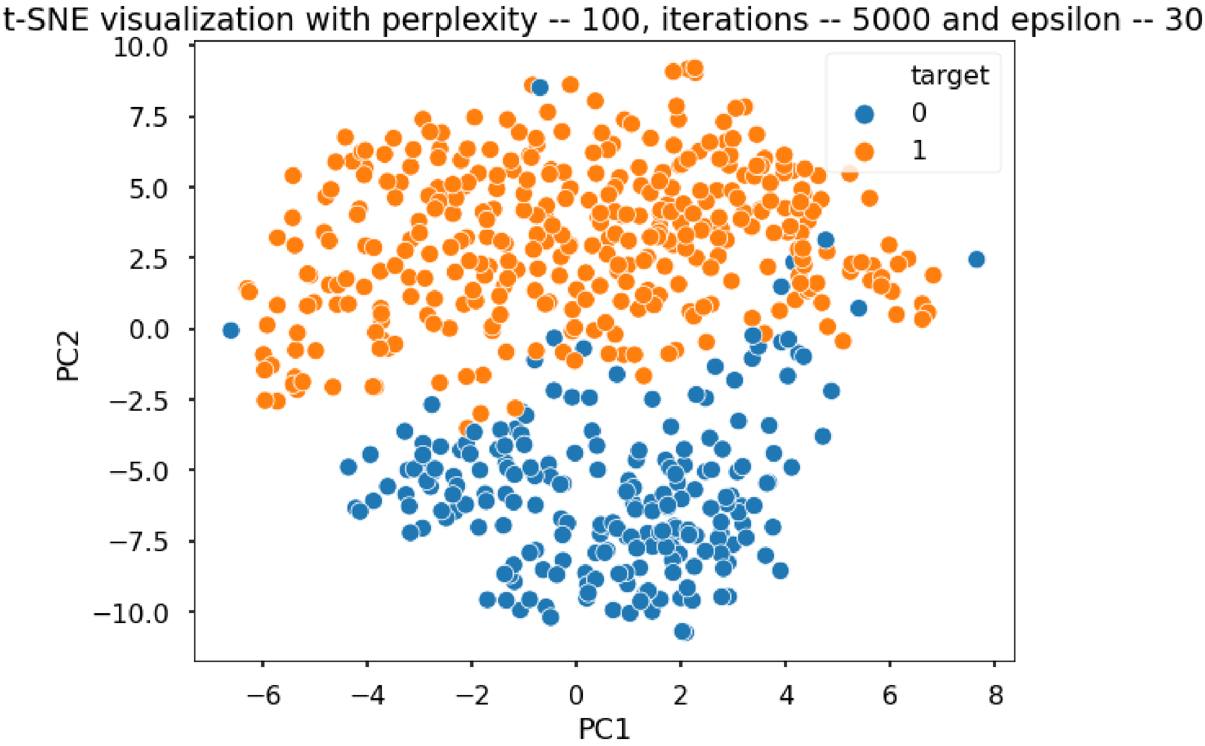

t-SNE visualization with perplexity -- 100, iterations -- 1000 and epsilon -- 30
target
0
1
8
6
4
2
0
−2
−4
−6
PC2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
PC1

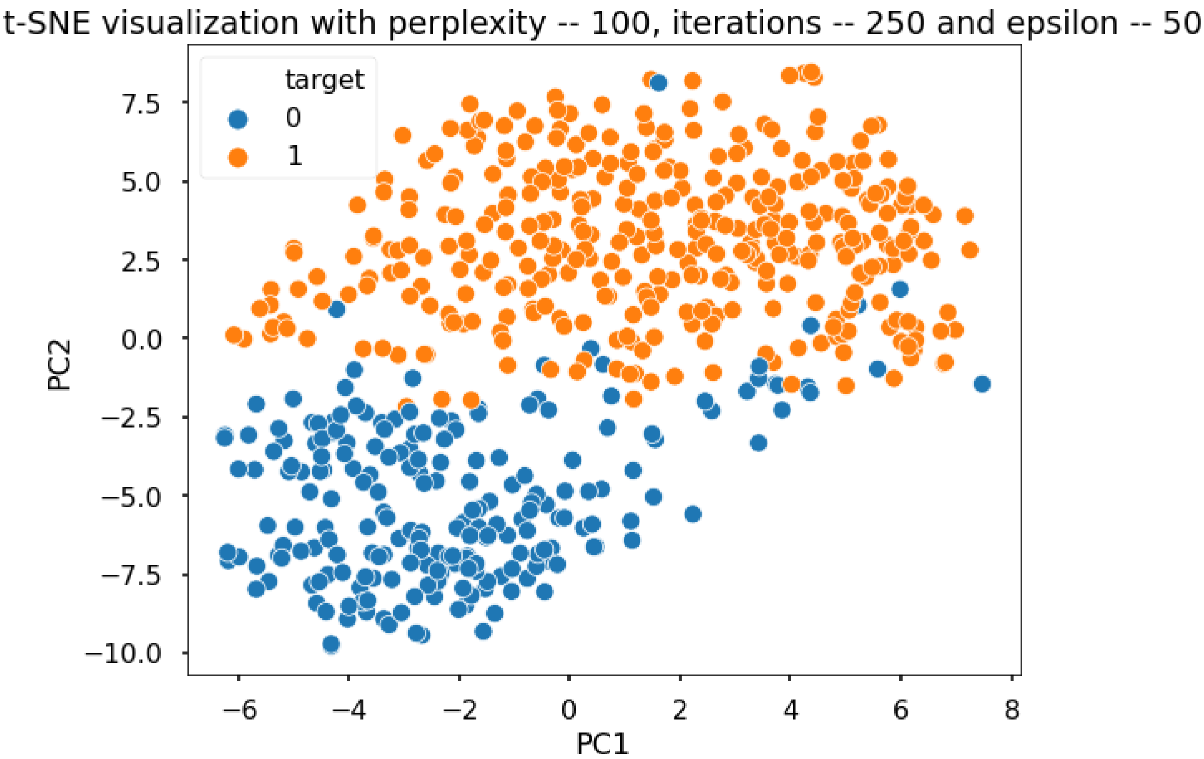
t-SNE visualization with perplexity -- 100, iterations -- 2000 and epsilon -- 30

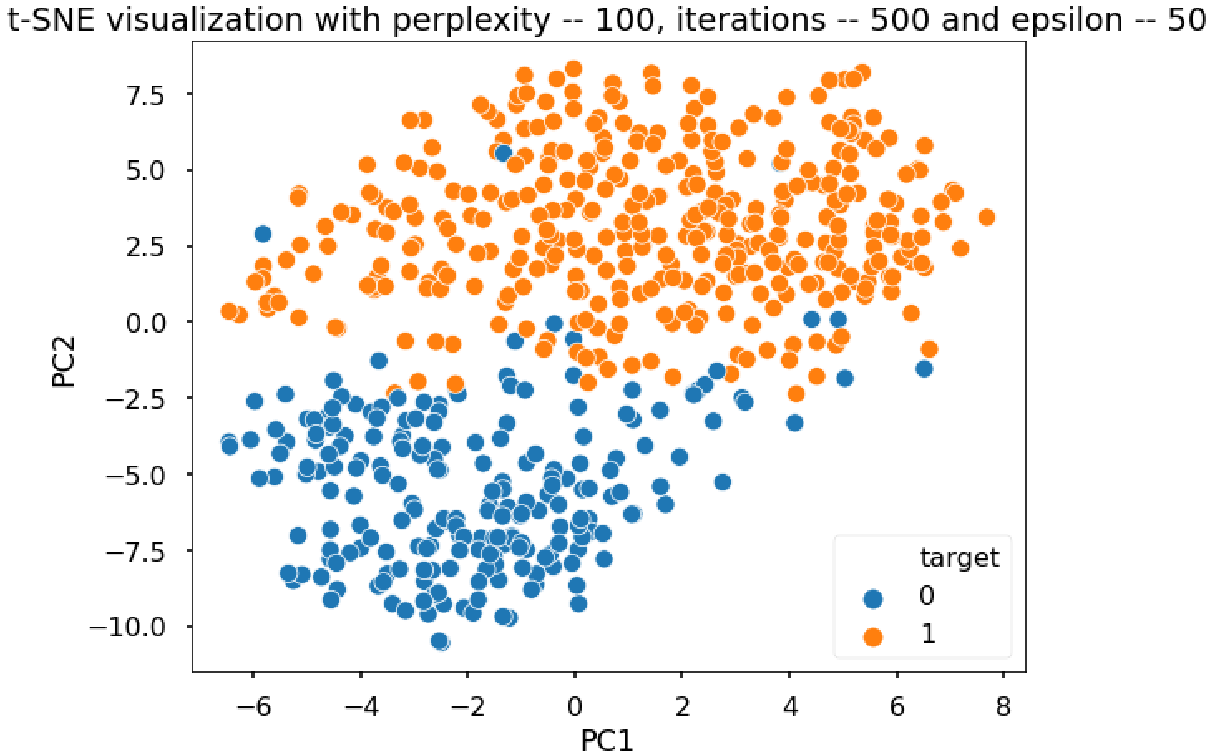


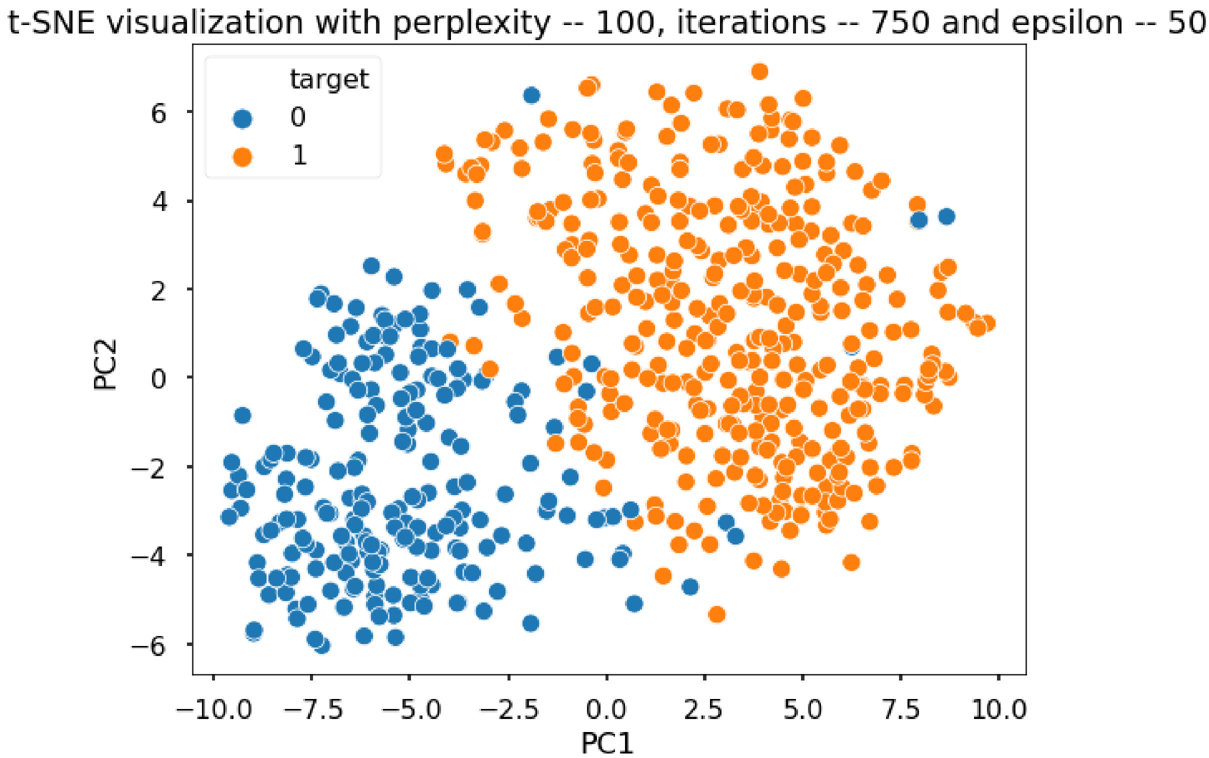
t-SNE visualization with perplexity -- 100, iterations -- 3000 and epsilon -- 30

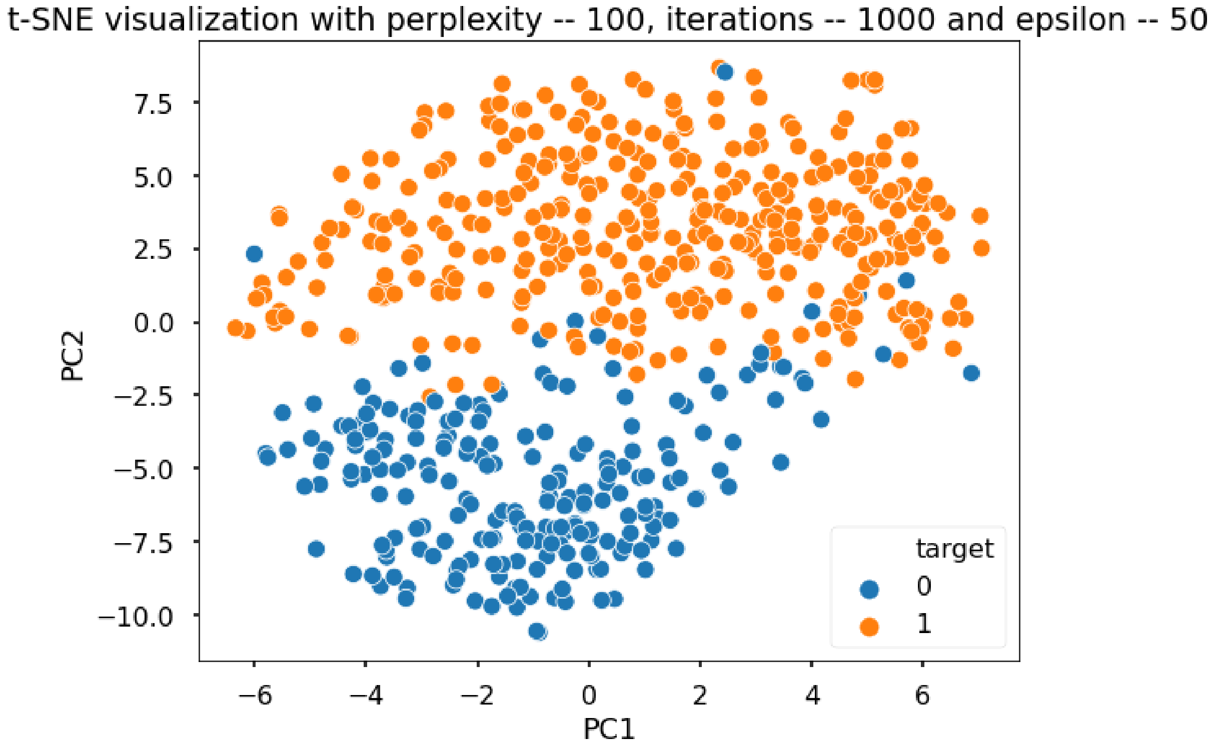


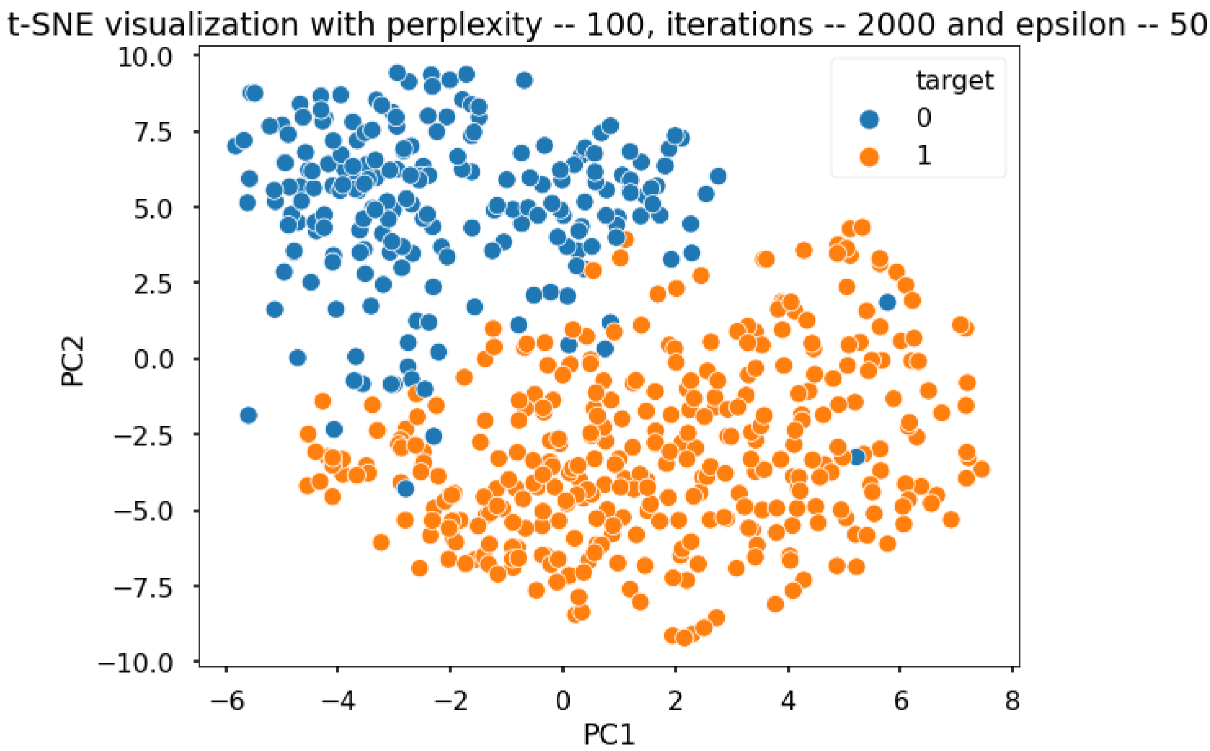

t-SNE visualization with perplexity -- 100, iterations -- 5000 and epsilon -- 30
target
0
1
PC2
PC1
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−6
−4
−2
0
2
4
6
8

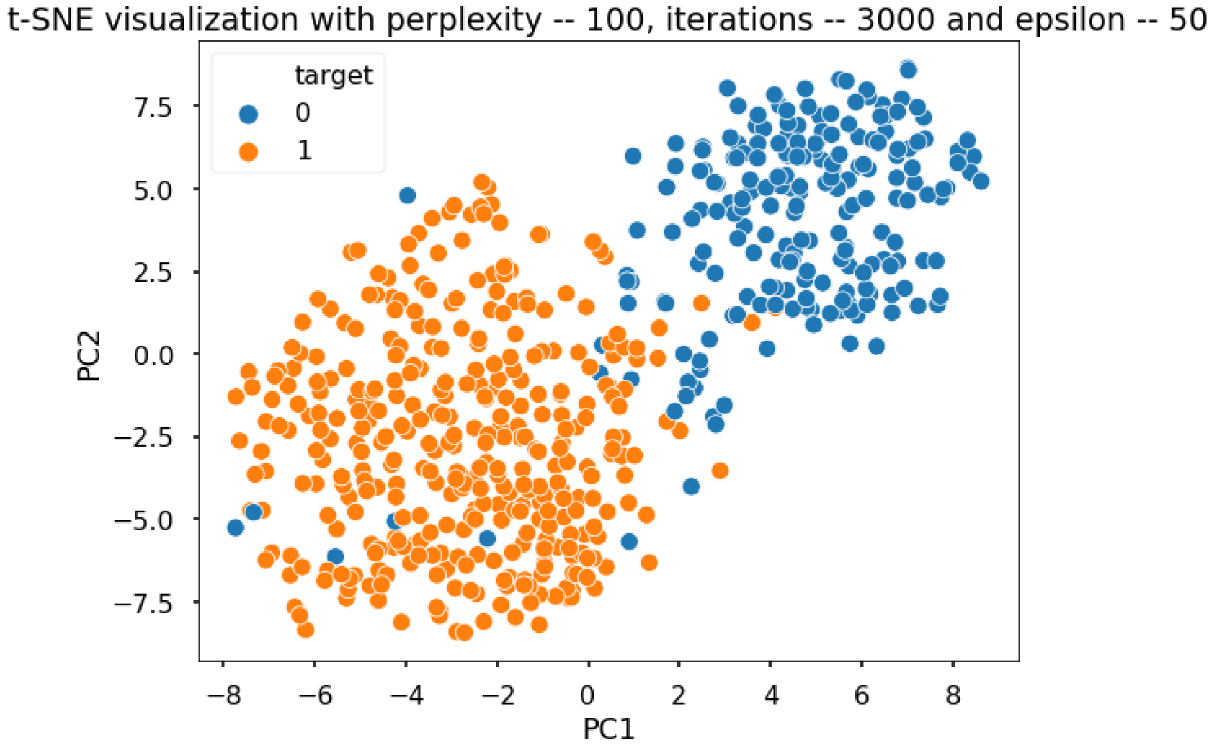

t-SNE visualization with perplexity -- 100, iterations -- 250 and epsilon -- 50
target
0
1
PC2
PC1
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−6
−4
−2
0
2
4
6
8

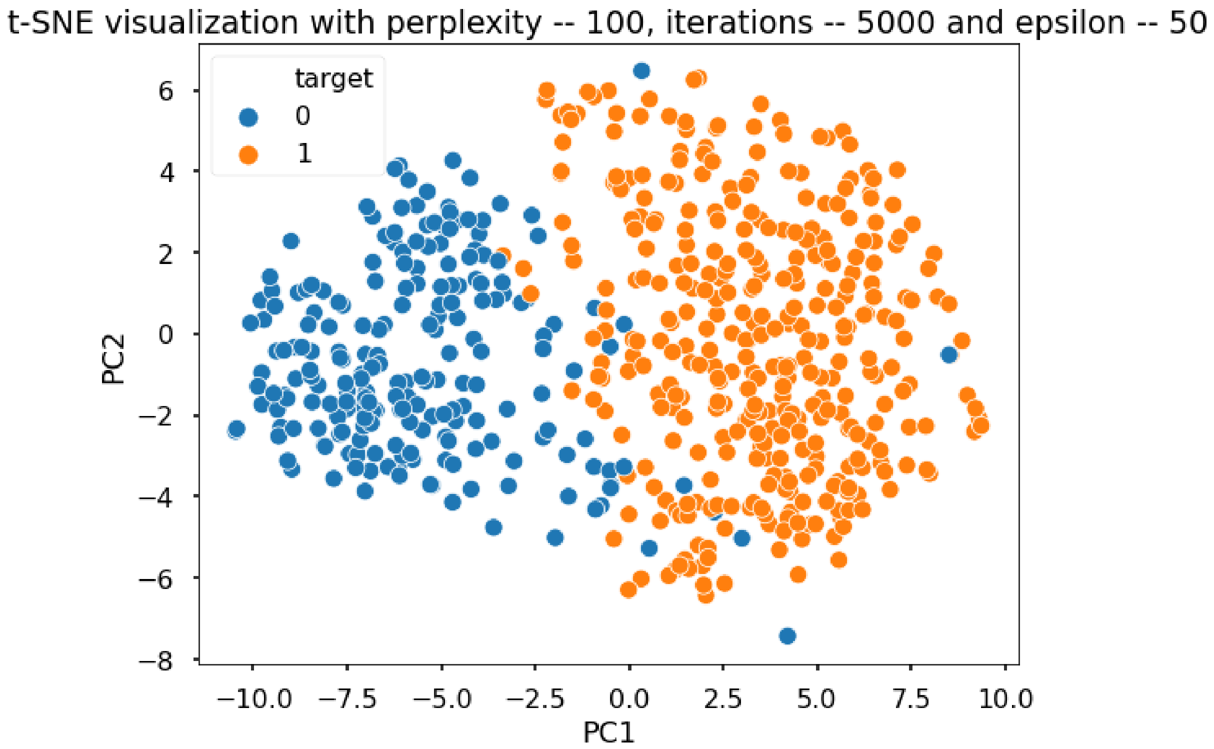

t-SNE visualization with perplexity -- 100, iterations -- 500 and epsilon -- 50
PC2
PC1
target
0
1
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−6
−4
−2
0
2
4
6
8


t-SNE visualization with perplexity -- 100, iterations -- 750 and epsilon -- 50
PC2
PC1
target
0
1
6
4
2
0
−2
−4
−6
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

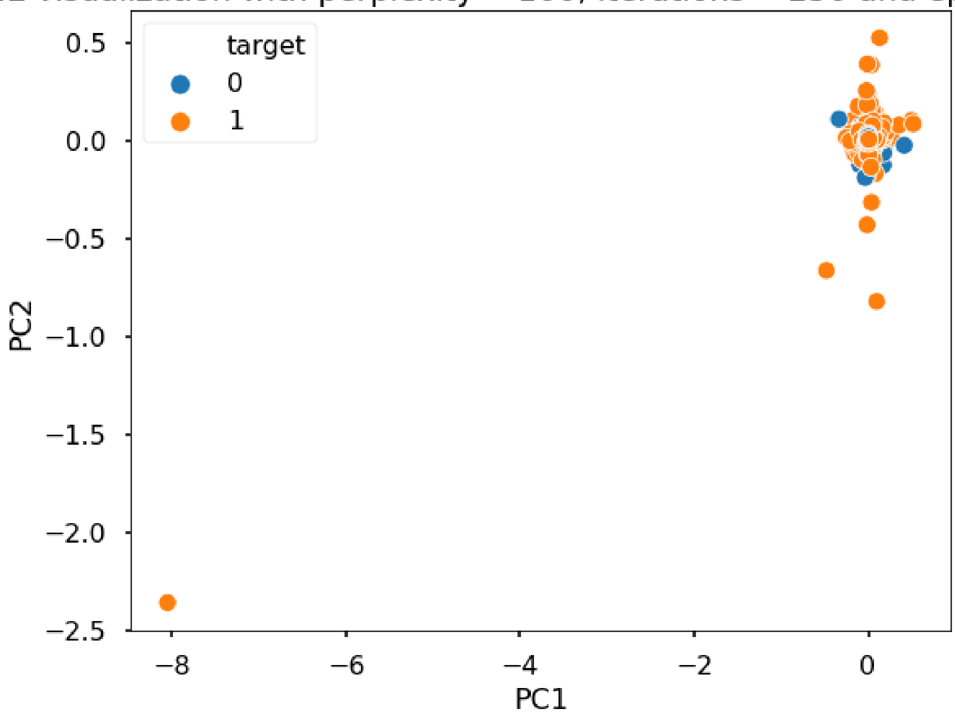

t-SNE visualization with perplexity -- 100, iterations -- 1000 and epsilon -- 50
PC2
PC1
target
0
1
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−6
−4
−2
0
2
4
6


t-SNE visualization with perplexity -- 100, iterations -- 2000 and epsilon -- 50
PC2
PC1
target
0
1
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−6
−4
−2
0
2
4
6
8

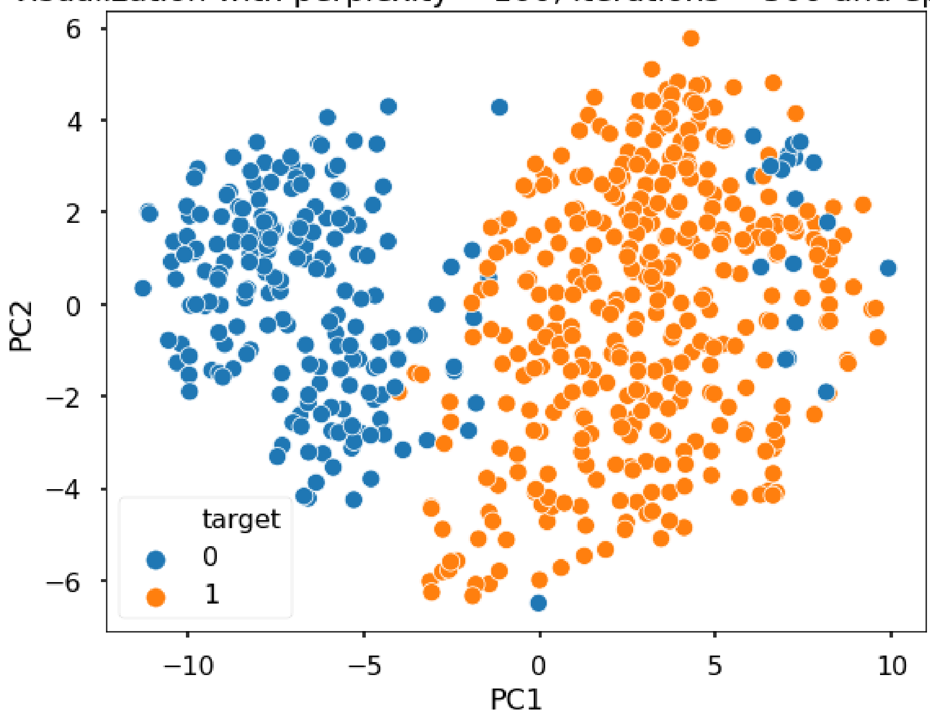

t-SNE visualization with perplexity -- 100, iterations -- 3000 and epsilon -- 50
target
0
1
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
PC2
−8
−6
−4
−2
0
2
4
6
8
PC1


t-SNE visualization with perplexity -- 100, iterations -- 5000 and epsilon -- 50
target
0
1
6
4
2
0
−2
−4
−6
−8
PC2
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0
PC1

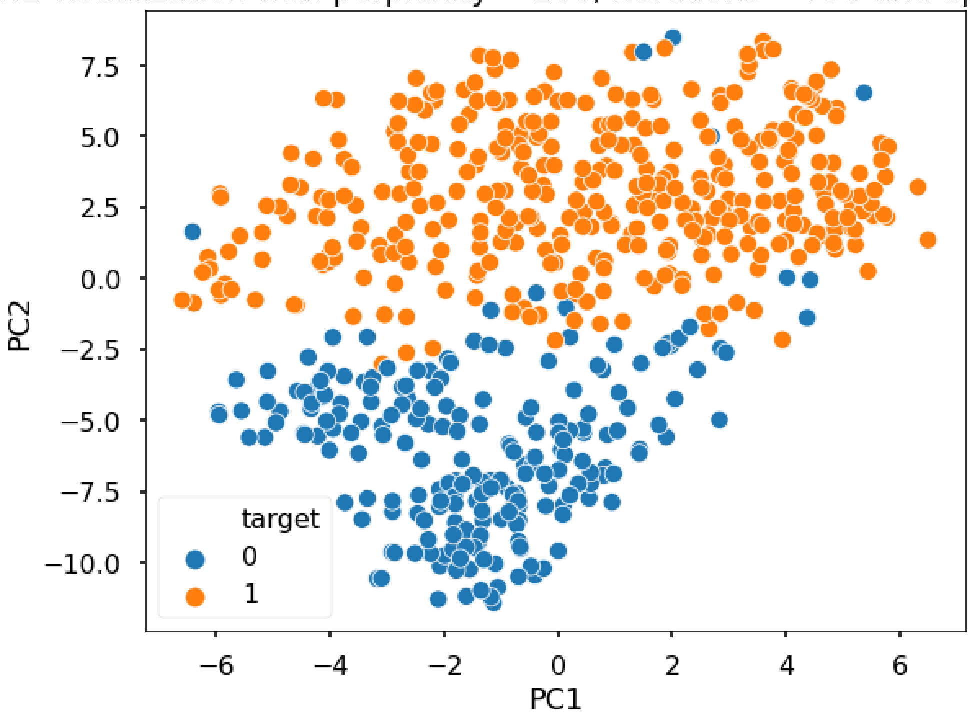
t-SNE visualization with perplexity -- 100, iterations -- 250 and epsilon -- 100



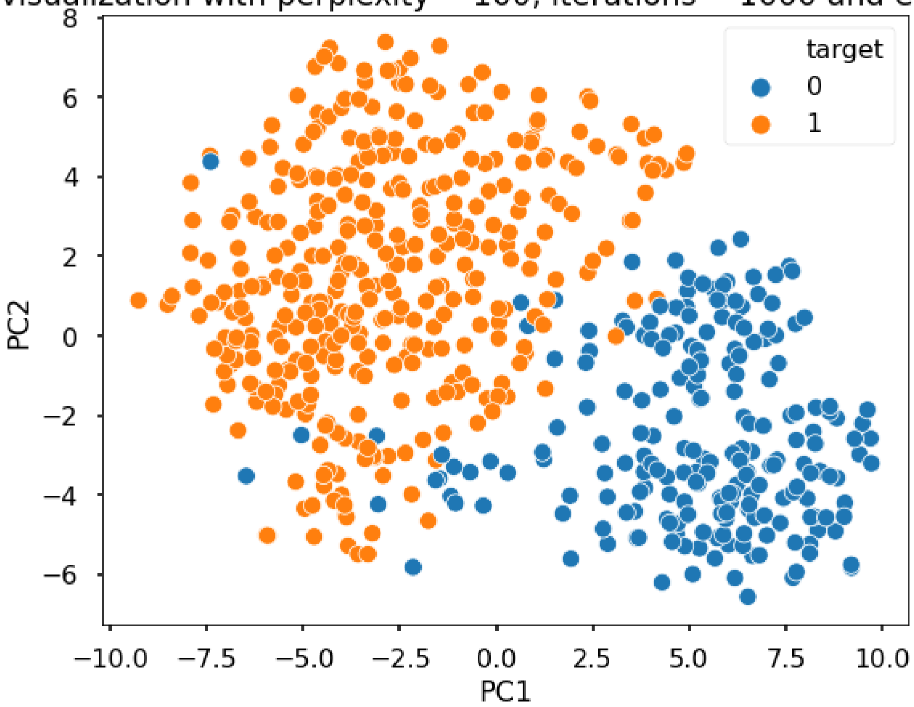
t-SNE visualization with perplexity -- 100, iterations -- 500 and epsilon -- 100

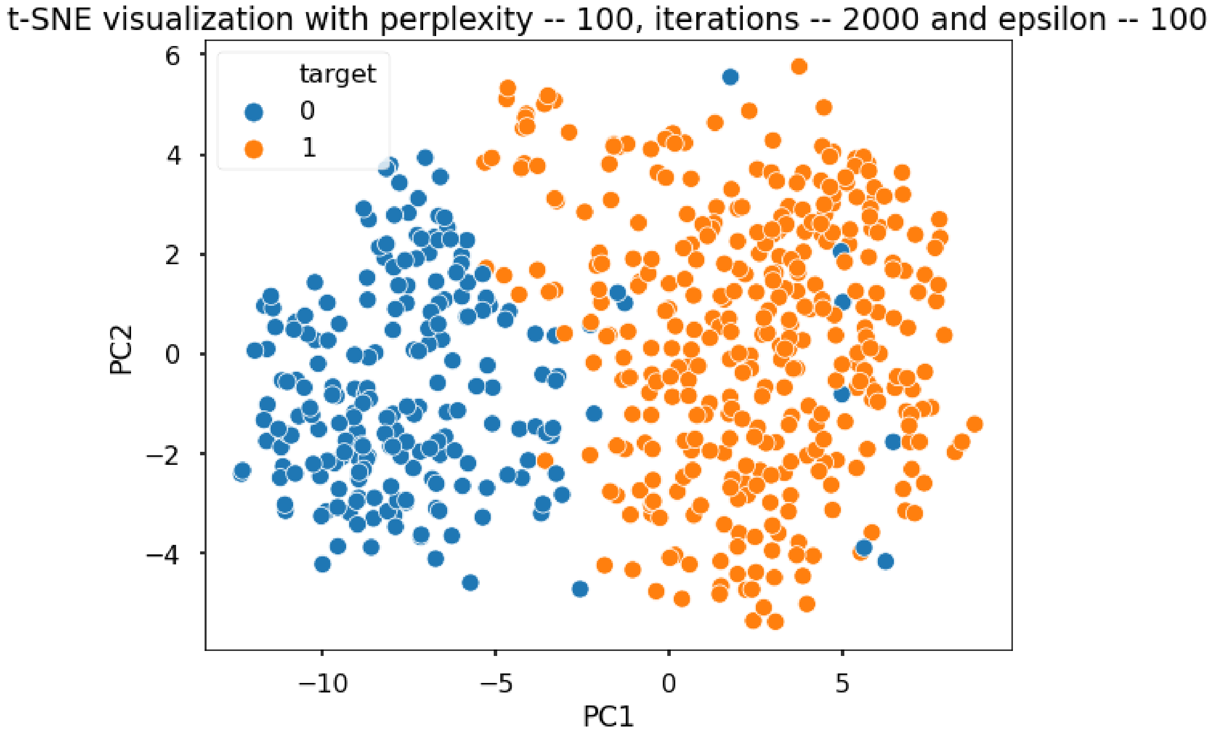


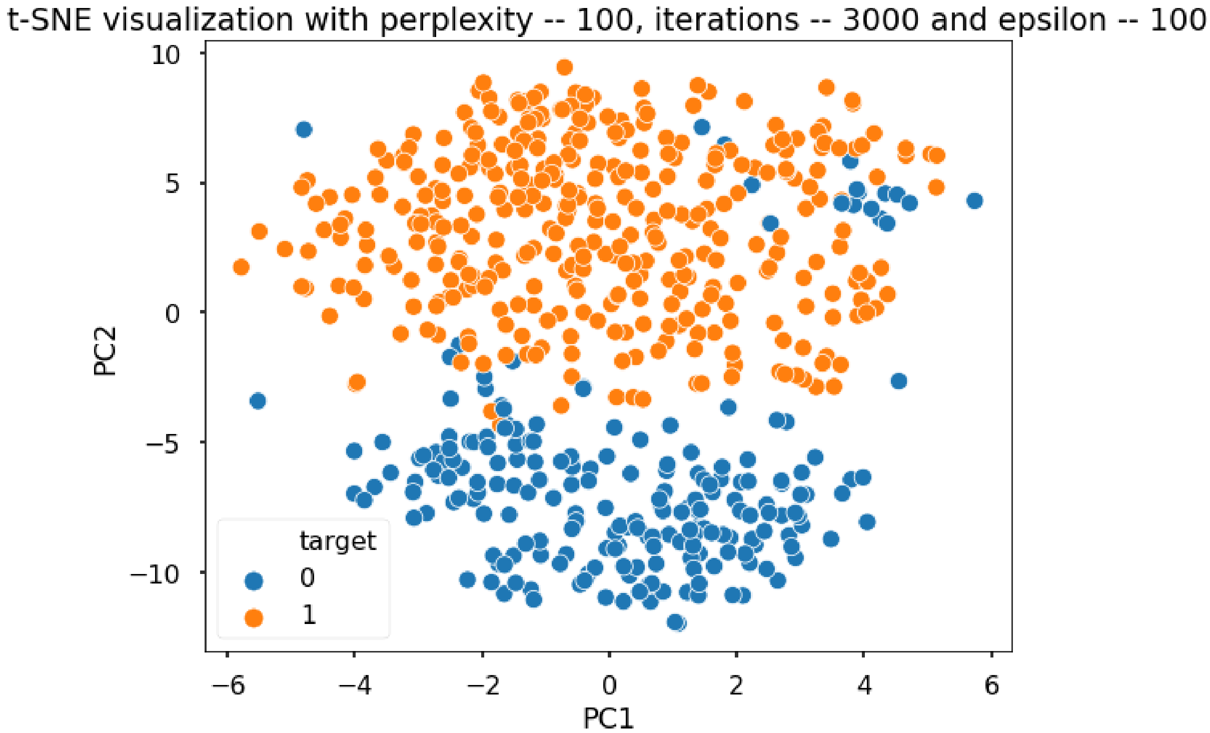
t-SNE visualization with perplexity -- 100, iterations -- 750 and epsilon -- 100

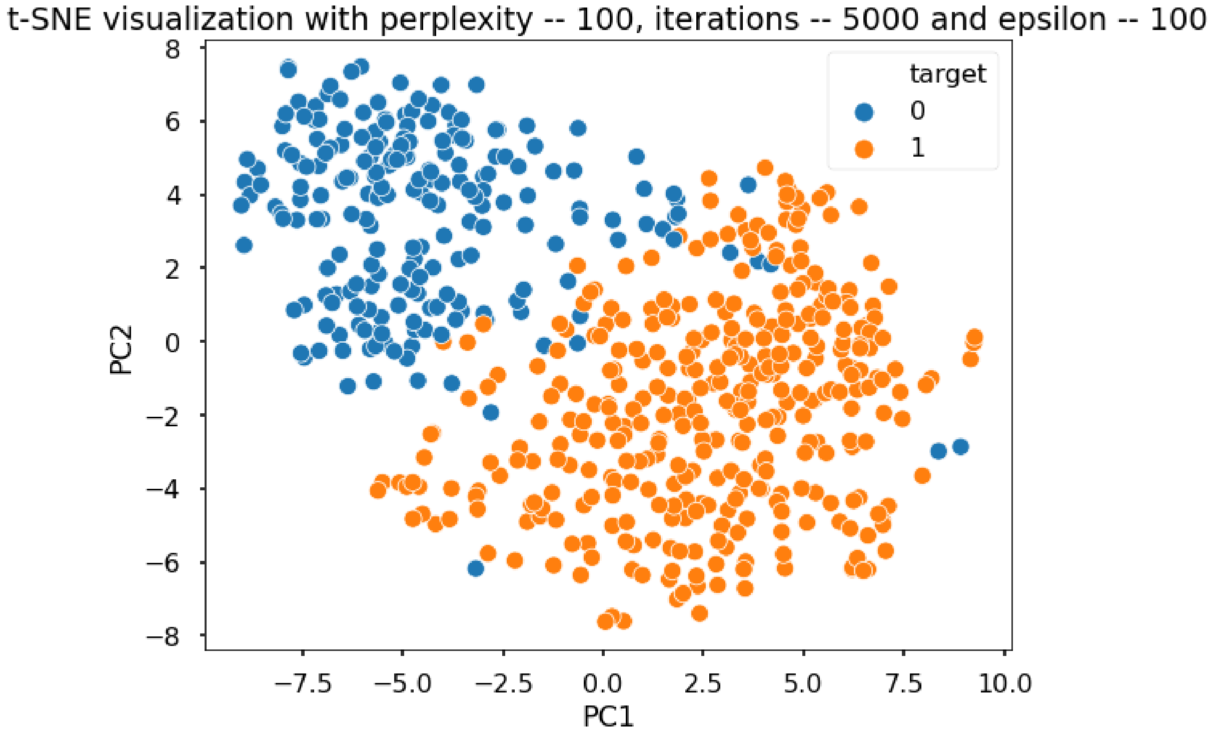


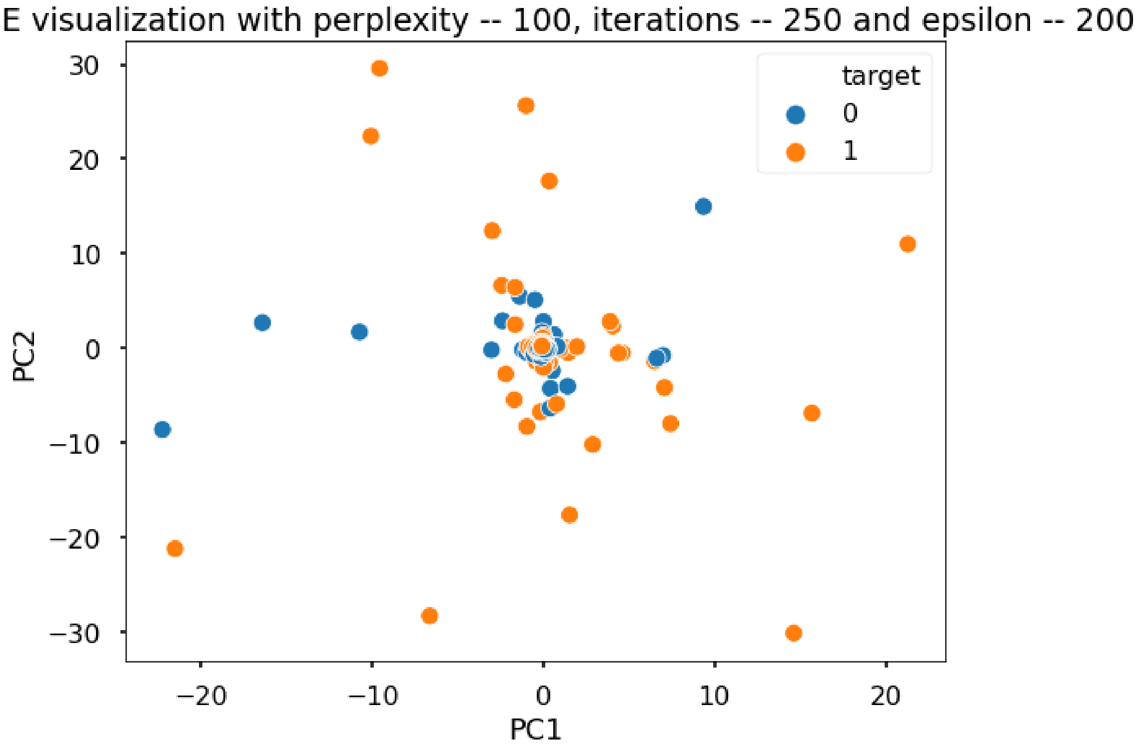
t-SNE visualization with perplexity -- 100, iterations -- 1000 and epsilon -- 100

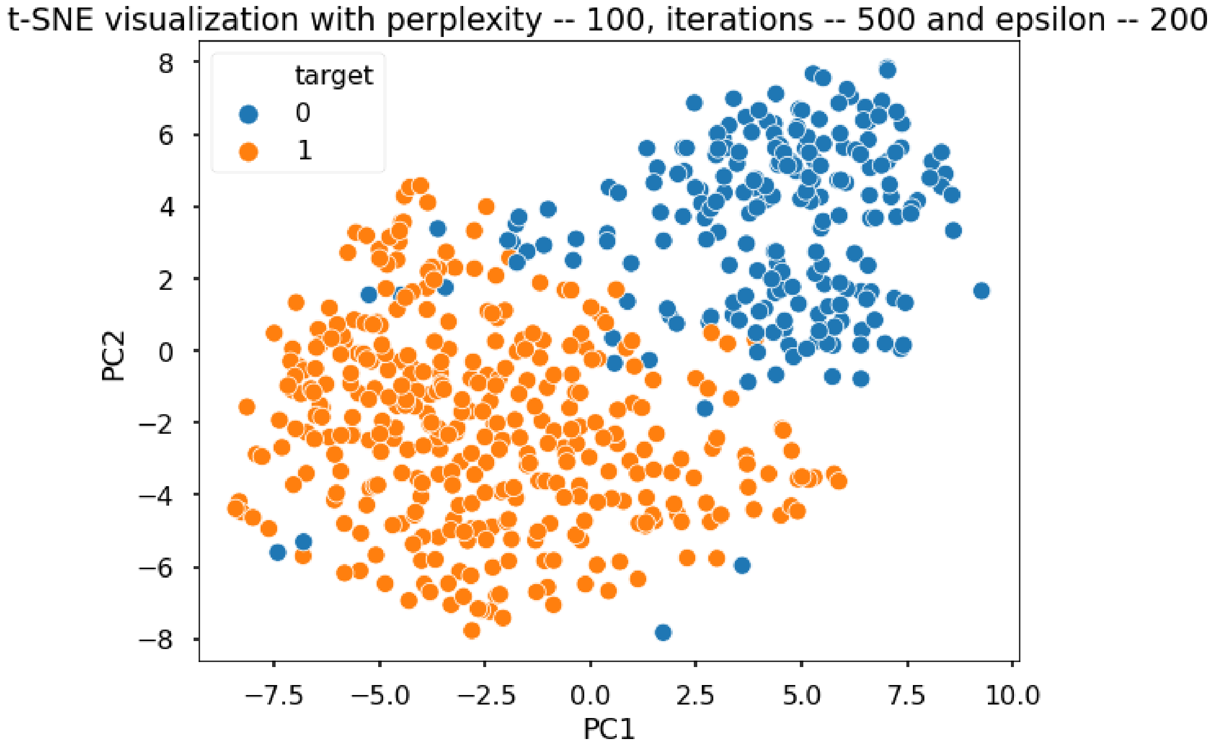


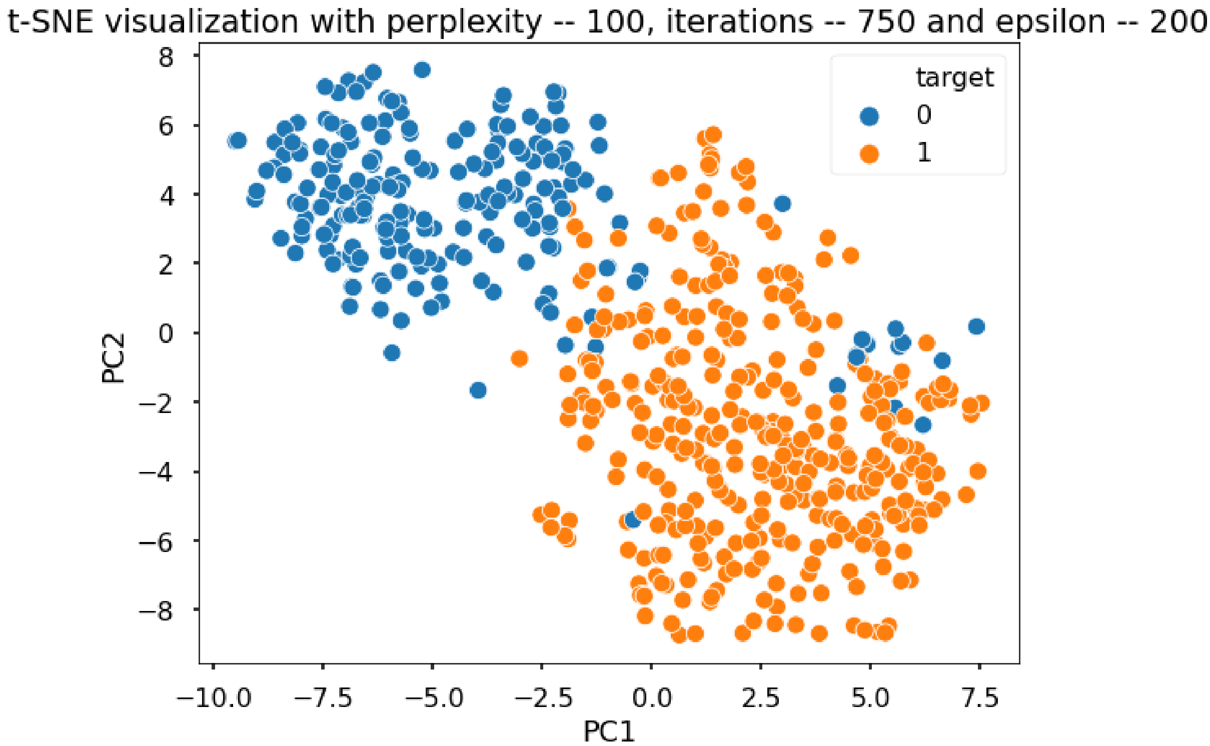

t-SNE visualization with perplexity -- 100, iterations -- 2000 and epsilon -- 100
target
0
1
PC2
PC1
6
4
2
0
−2
−4
−10
−5
0
5

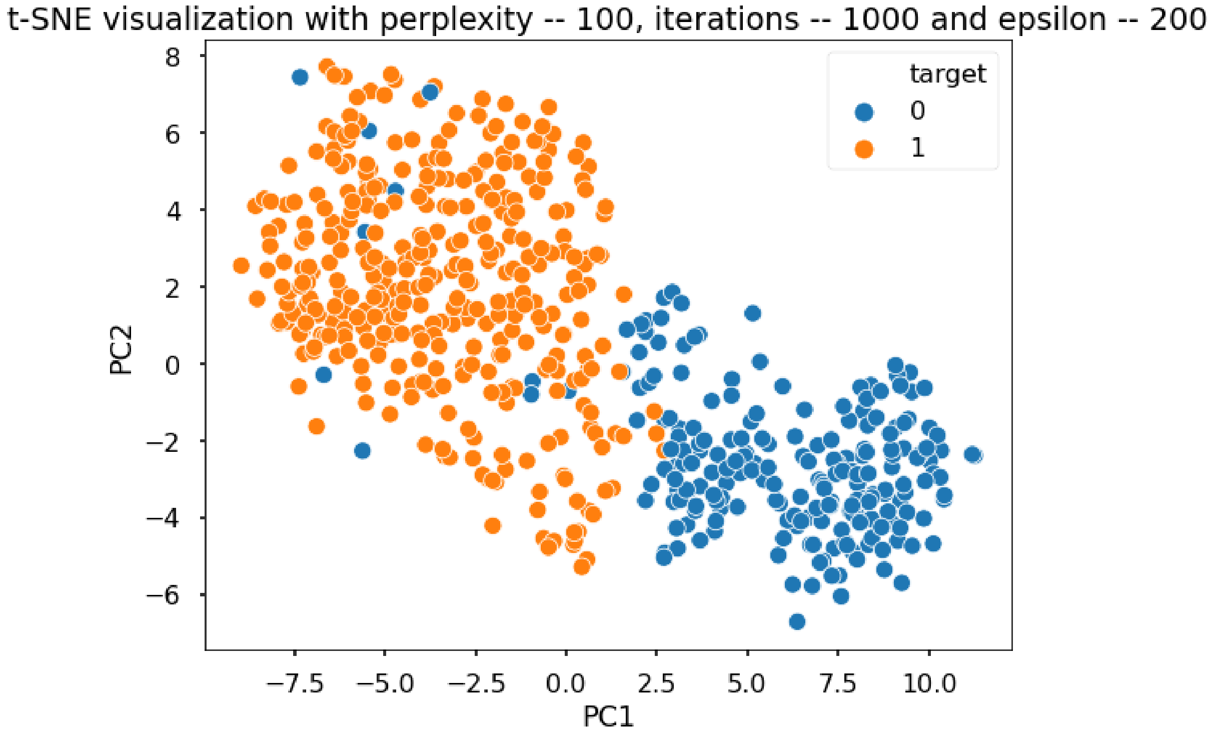

t-SNE visualization with perplexity -- 100, iterations -- 3000 and epsilon -- 100
target
0
1
PC2
PC1
10
5
0
−5
−10
−6
−4
−2
0
2
4
6


t-SNE visualization with perplexity -- 100, iterations -- 5000 and epsilon -- 100
PC1
PC2
target
0
1

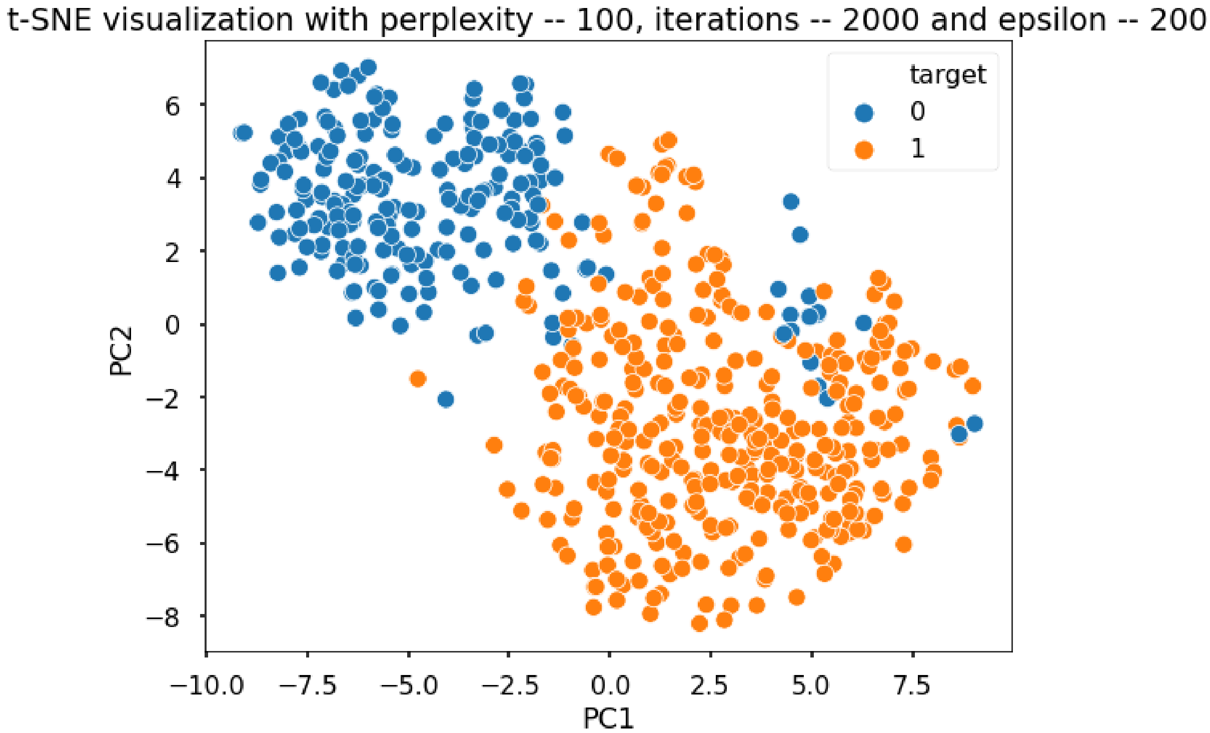

t-SNE visualization with perplexity -- 100, iterations -- 250 and epsilon -- 200
PC1
PC2
target
0
1


t-SNE visualization with perplexity -- 100, iterations -- 500 and epsilon -- 200
target
0
1
PC2
PC1
8
6
4
2
0
−2
−4
−6
−8
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

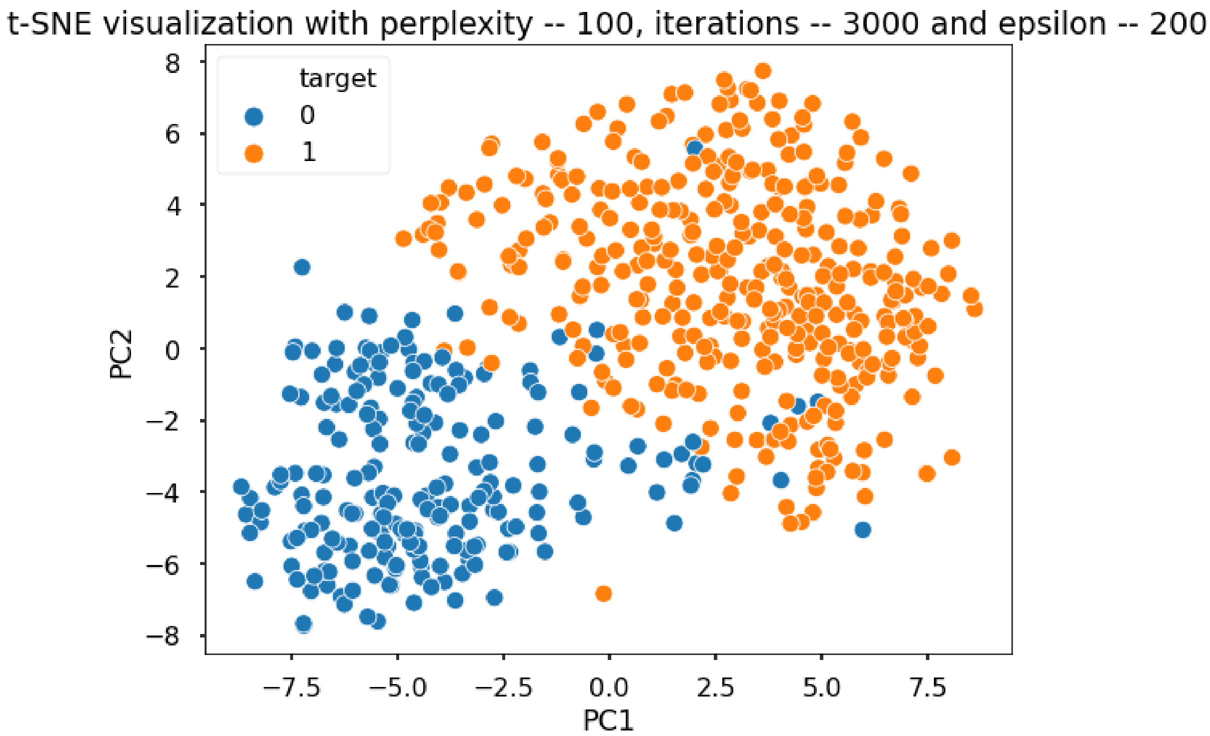

t-SNE visualization with perplexity -- 100, iterations -- 750 and epsilon -- 200
target
0
1
PC2
PC1
8
6
4
2
0
−2
−4
−6
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5

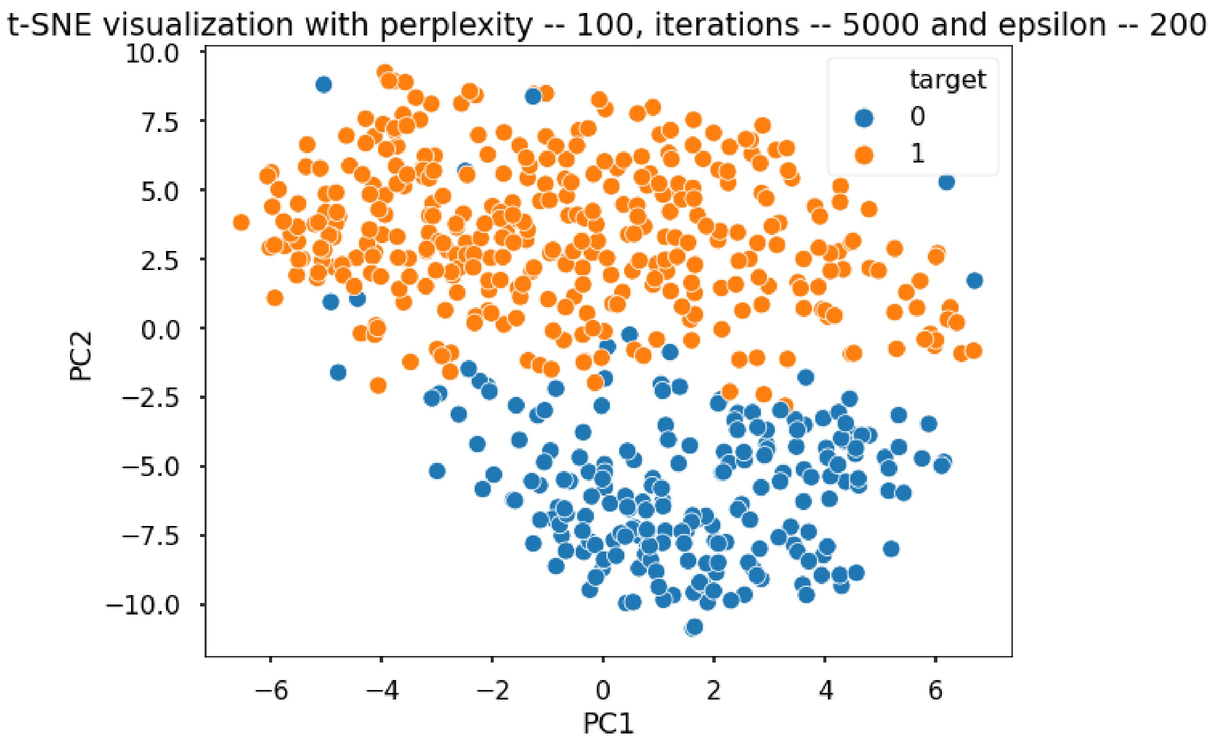


t-SNE visualization with perplexity -- 100, iterations -- 1000 and epsilon -- 200
target
0
1
PC2
PC1
8
6
4
2
0
−2
−4
−6
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0



t-SNE visualization with perplexity -- 100, iterations -- 2000 and epsilon -- 200
target
0
1
PC2
PC1
6
4
2
0
−2
−4
−6
−8
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5





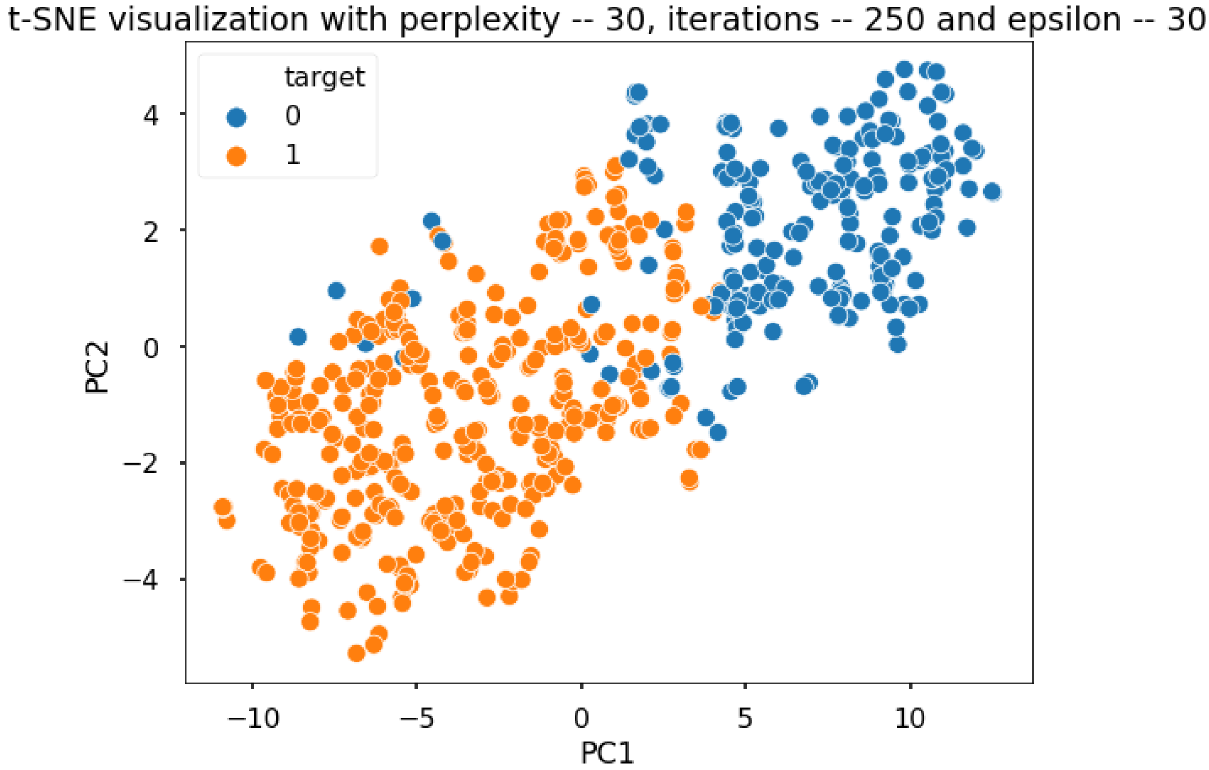
In the above plots, my obervation is that with Perplexity=30 and Epsilon=30 gives us the better separation of classes with a very few outliers. However, in the other plots the outliers are quite high and they are largely distant from the respective class cluster.

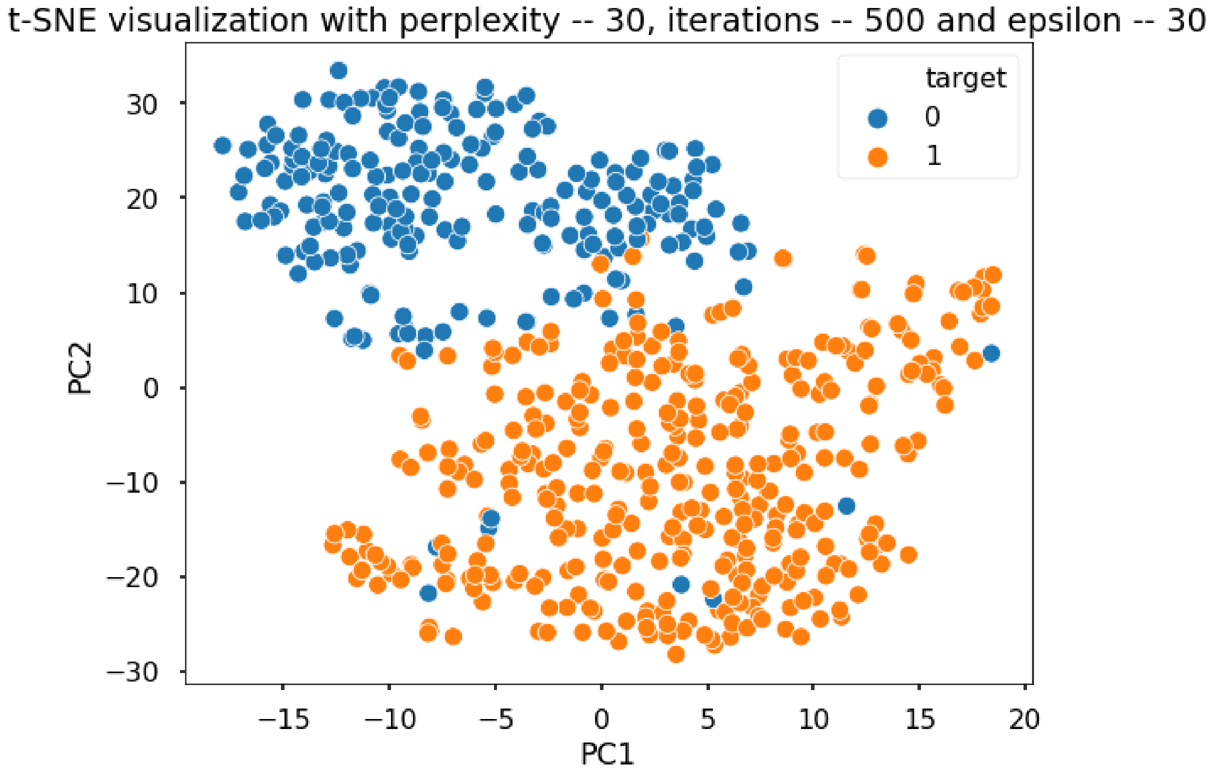
So, next step is to run the T-SNE for this combination multiple times with various iterations.

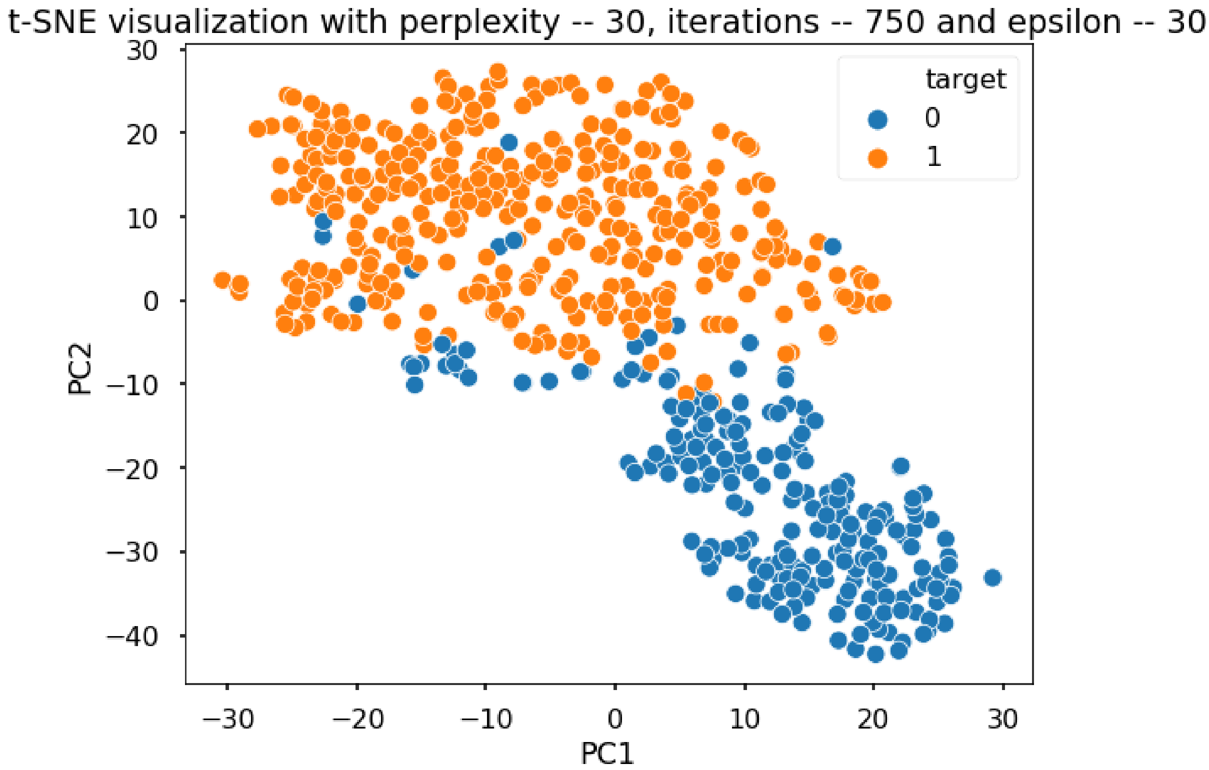
CASE-II-Multiple_runs

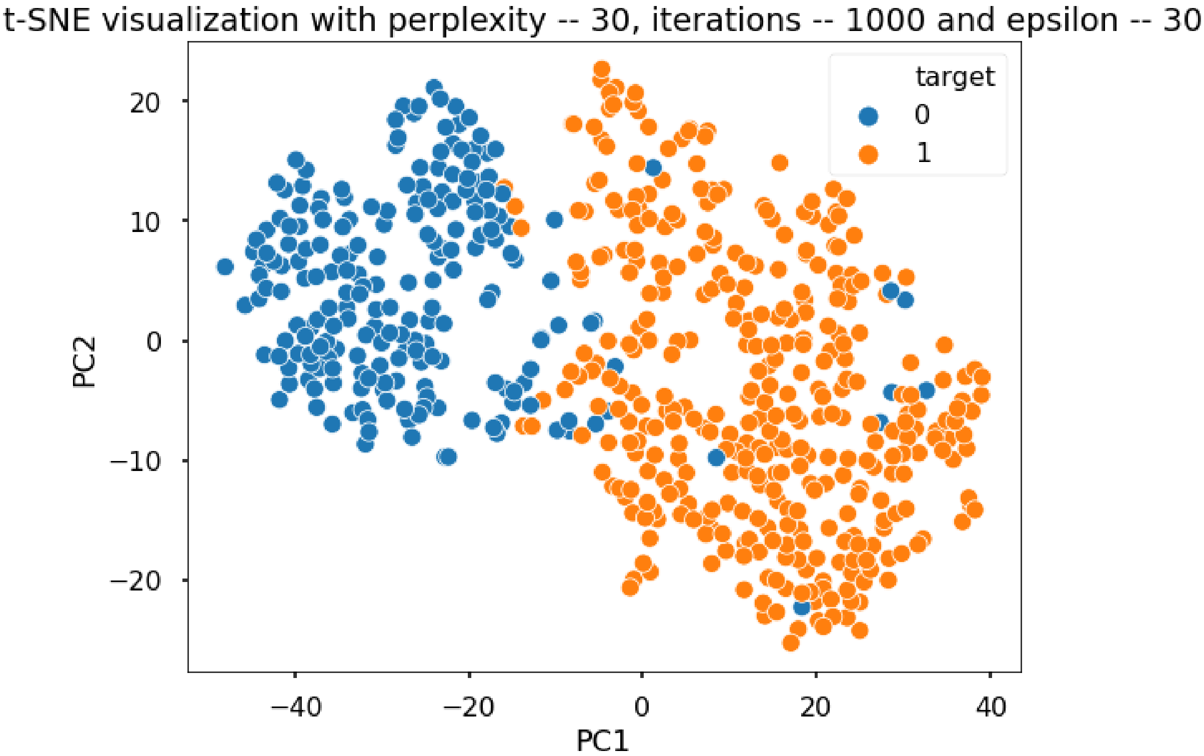
Running t-SNE for a range of iterations with learning rate and perplexity as 30 and embedding method as 'random'

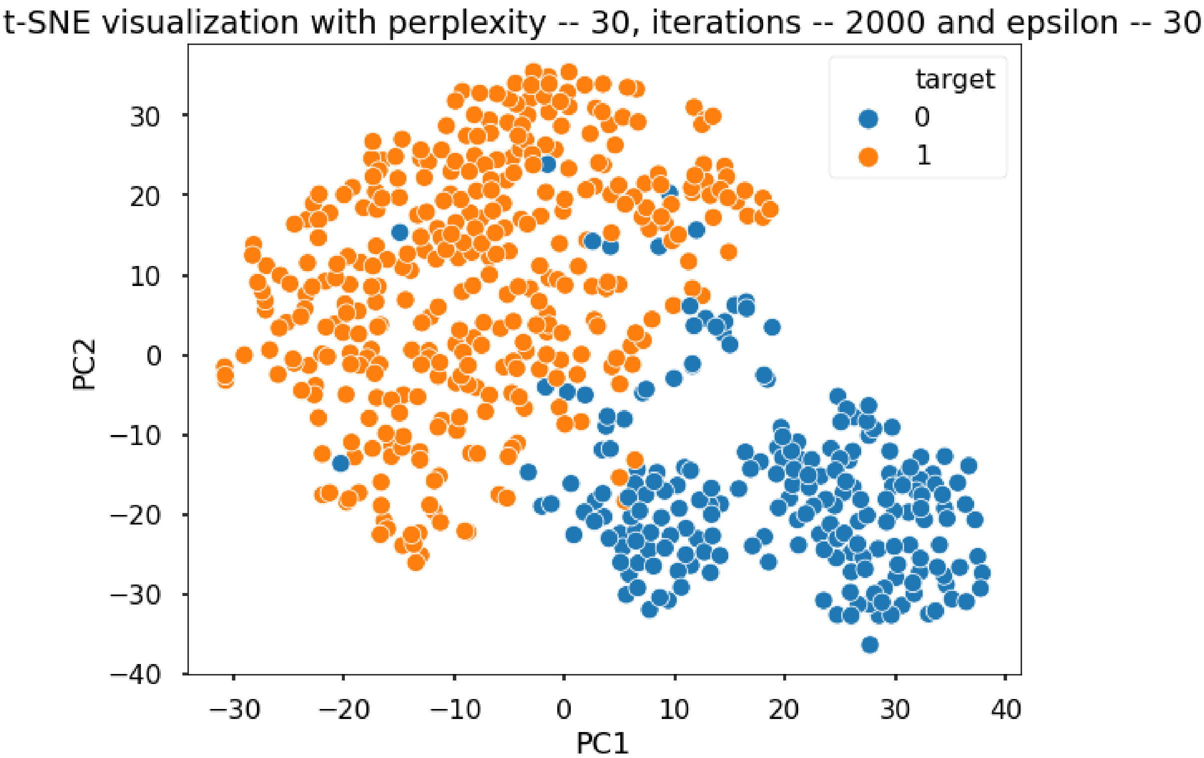
In [7]:
```
iterations = [250,500,750,1000,2000,3500,5000]

for idx in range(len(iterations)):
    tsne1 = TSNE(n_components=2,init='random',perplexity=30,learning_rate=30,n_iter=
                 method='exact',
                 n_jobs=-1)
    cancer_tsne_pcmps = pd.DataFrame(tsne1.fit_transform(cancer_norm_df),columns=['P
    cancer_tsne_pcmps = pd.concat([cancer_tsne_pcmps,cancer_df['target']],axis=1)
    with plt.style.context('seaborn-poster'):
        plt.figure(figsize=(9,7))
        sns.scatterplot(data=cancer_tsne_pcmps,x='PC1',y='PC2',hue='target')
        plt.title("t-SNE visualization with perplexity -- {0}, iterations -- {1} and


        plt.show()
```

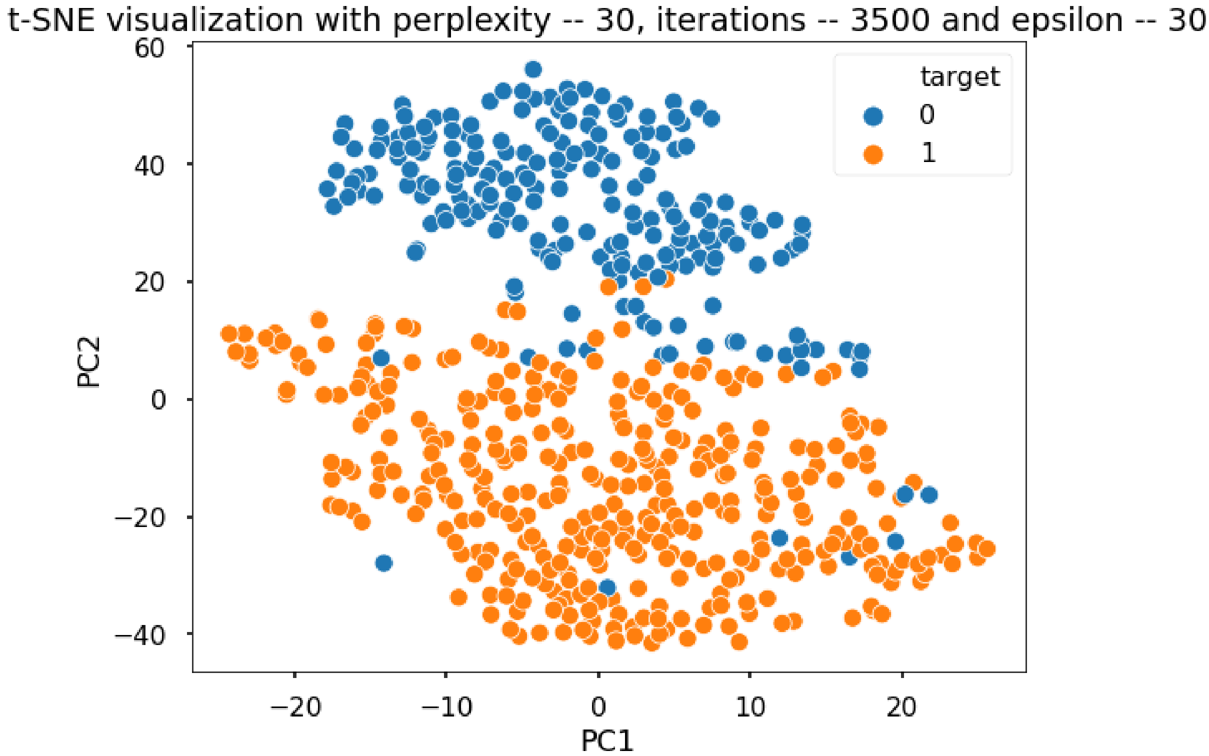


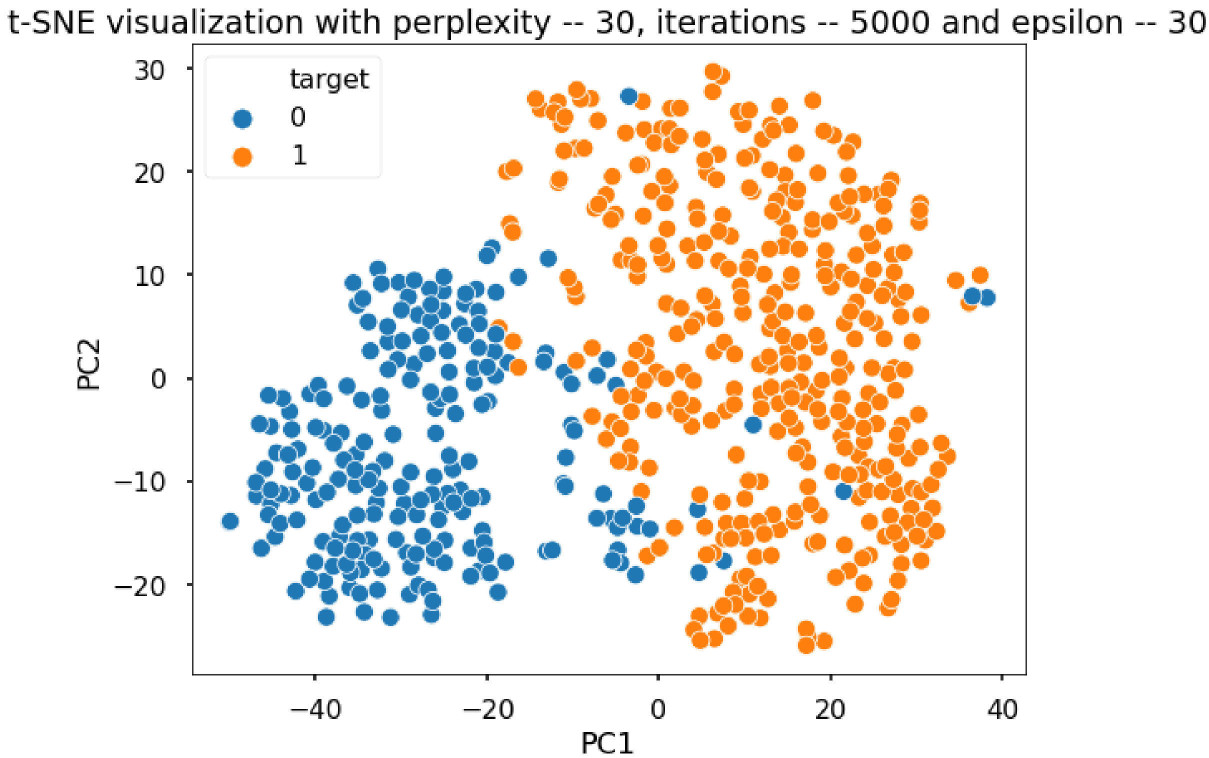

t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 30
target
0
1
PC2
PC1
30
20
10
0
−10
−20
−30
−15
−10
−5
0
5
10
15
20


t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 30
target
0
1
PC2
PC1
30
20
10
0
−10
−20
−30
−40
−30
−20
−10
0
10
20
30


t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 30
target
0
1
20
10
0
−10
−20
PC2
−40
−20
0
20
40
PC1


t-SNE visualization with perplexity -- 30, iterations -- 2000 and epsilon -- 30
target
0
1
30
20
10
0
−10
−20
−30
−40
PC2
−30
−20
−10
0
10
20
30
40
PC1



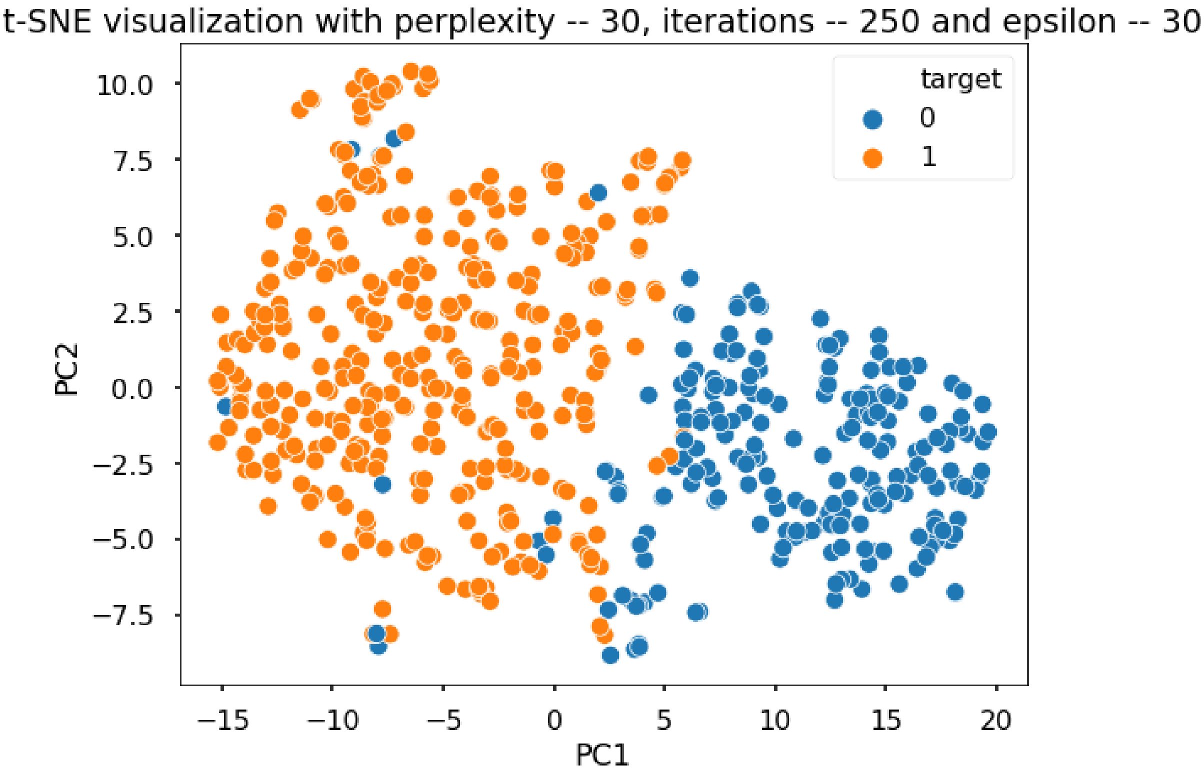


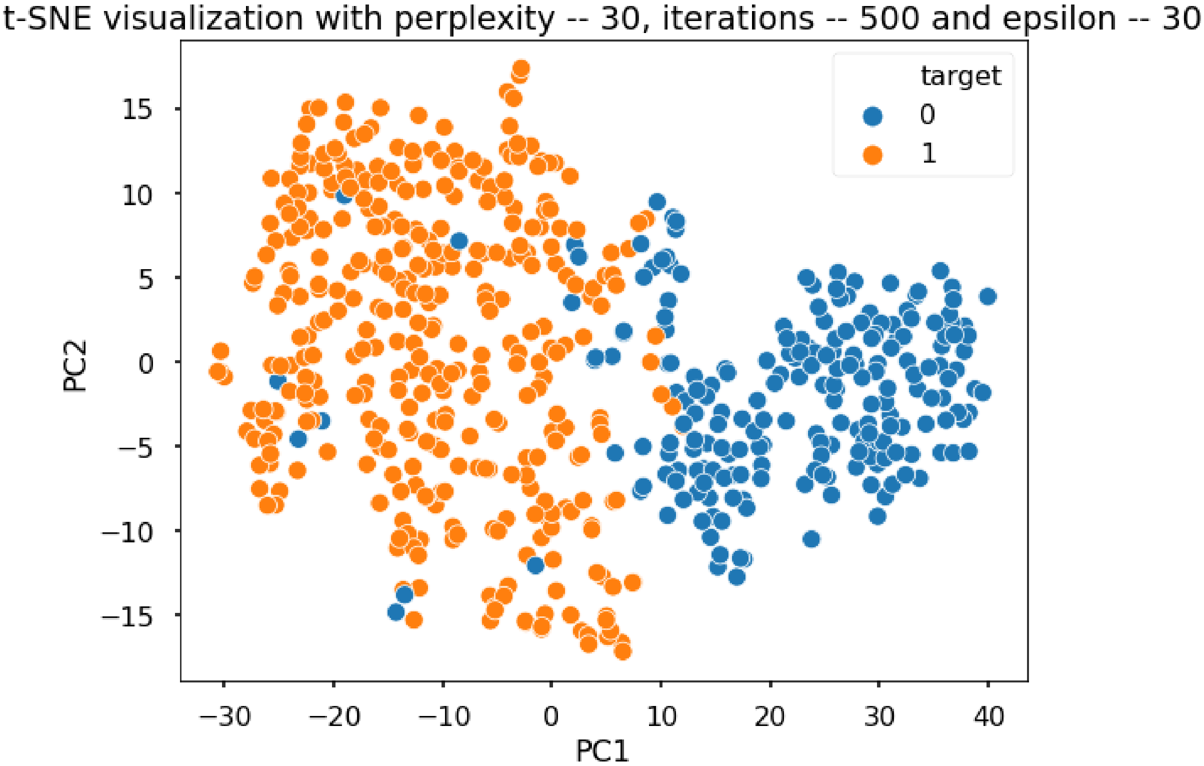
CASE-III-Multiple_runs

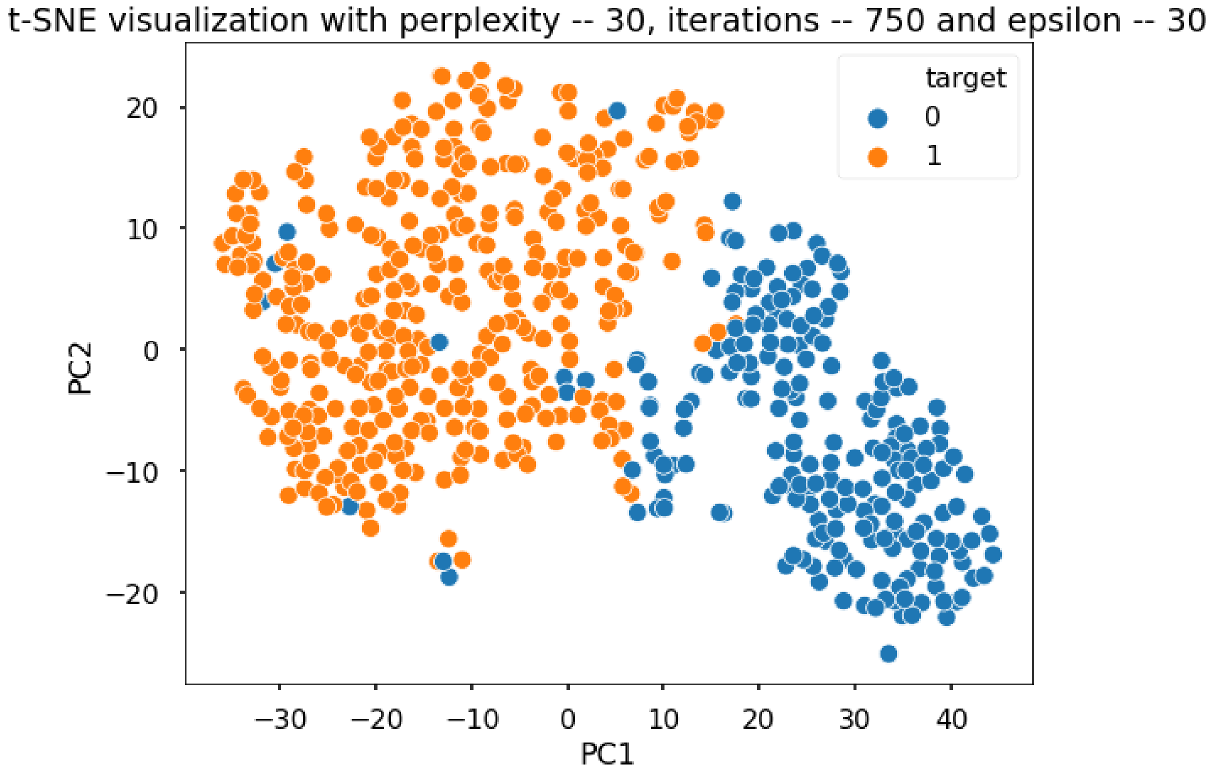
Running t-SNE for a range of iterations with learning rate and perplexity as 30 and embedding method as 'PCA'

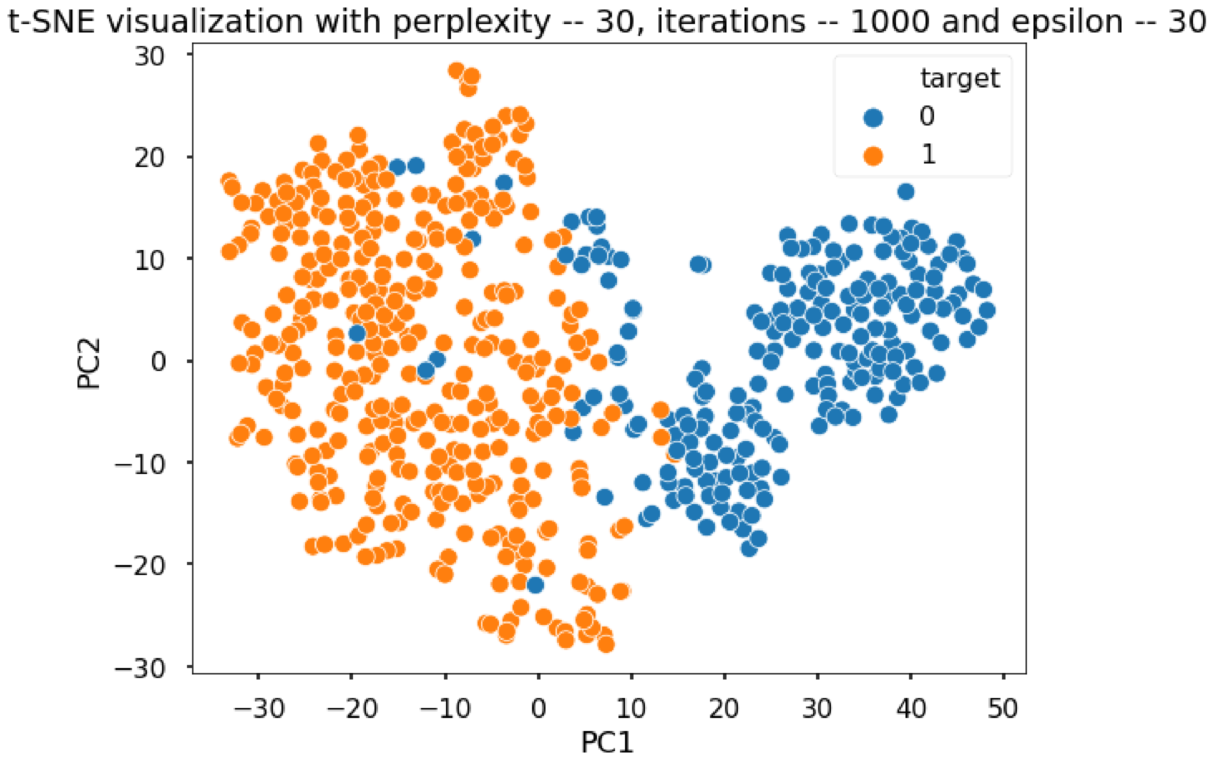
In [8]:
```
iterations = [250,500,750,1000,2000,3500,5000]

for idx in range(len(iterations)):
    tsne1 = TSNE(n_components=2,init='pca',perplexity=30,learning_rate=30,n_iter=ite
                 method='exact',
```

```
                     n_jobs=-1)
    cancer_tsne_pcmps = pd.DataFrame(tsne1.fit_transform(cancer_norm_df),columns=['P
    cancer_tsne_pcmps = pd.concat([cancer_tsne_pcmps,cancer_df['target']],axis=1)
    with plt.style.context('seaborn-poster'):
        plt.figure(figsize=(9,7))
        sns.scatterplot(data=cancer_tsne_pcmps,x='PC1',y='PC2',hue='target')
        plt.title("t-SNE visualization with perplexity -- {0}, iterations -- {1} and


        plt.show()
```

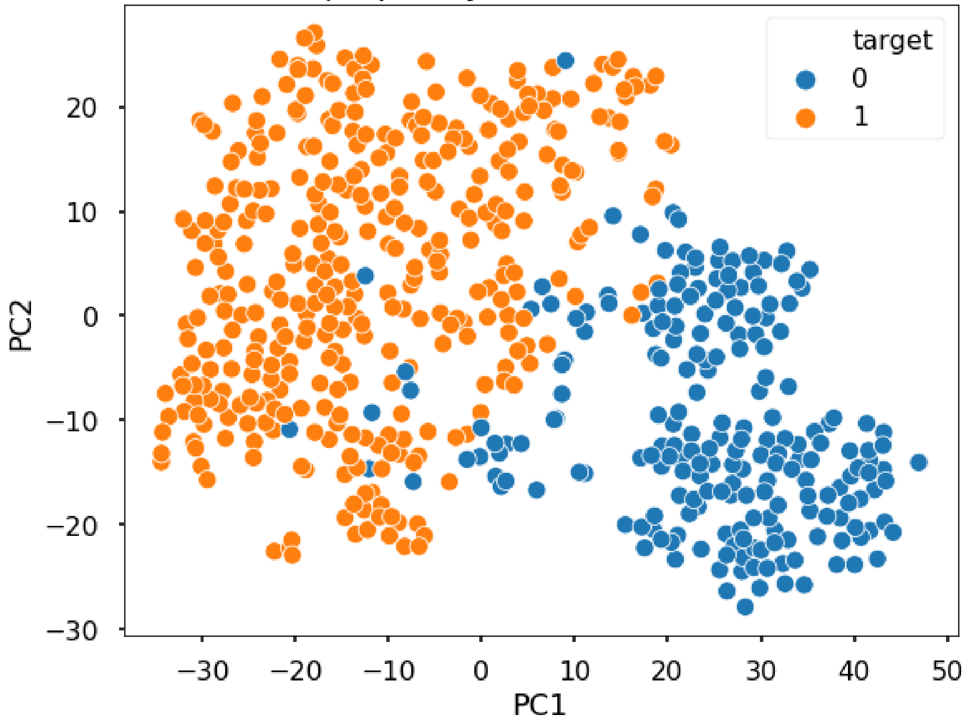




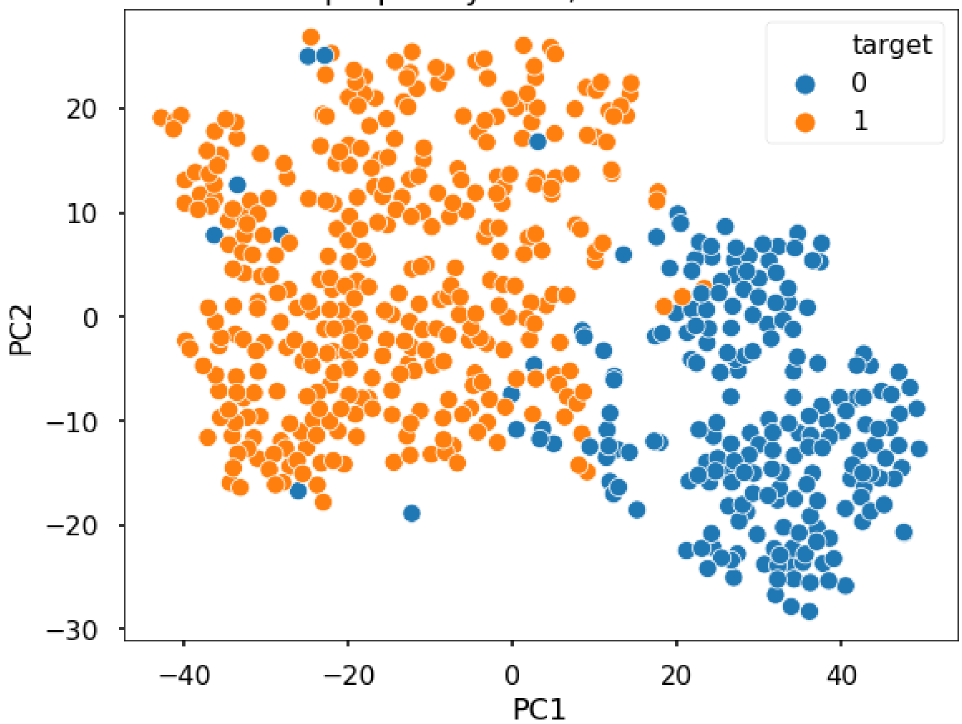

t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 30
target
0
1
20
10
0
−10
−20
PC2
−30
−20
−10
0
10
20
30
40
PC1

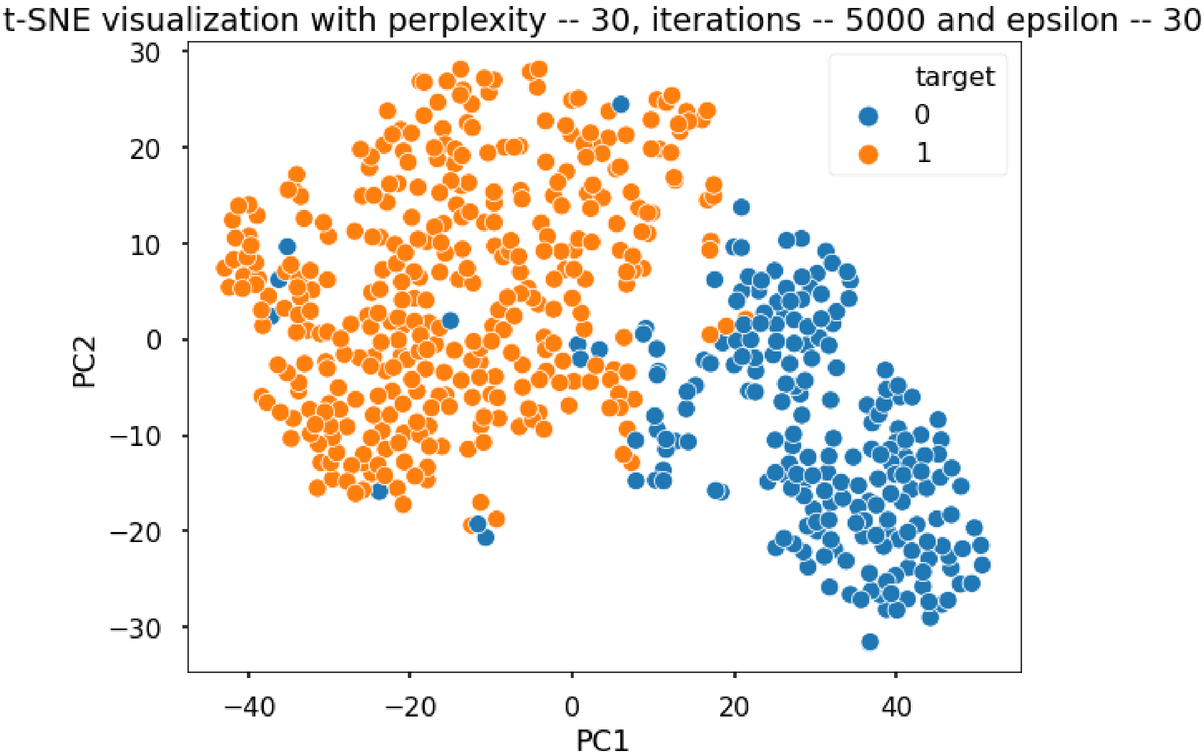

t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 30
target
0
1
30
20
10
0
−10
−20
−30
PC2
−30
−20
−10
0
10
20
30
40
50
PC1

t-SNE visualization with perplexity -- 30, iterations -- 2000 and epsilon -- 30



t-SNE visualization with perplexity -- 30, iterations -- 3500 and epsilon -- 30





IRIS_Dataset

In [26]:
```
iris_df = pd.DataFrame(iris.data,columns=iris.feature_names)
iris_stand_df = pd.concat([pd.DataFrame(ss.fit_transform(iris_df),columns=iris.featu
                           pd.DataFrame(iris.target,columns=['target'])],axis=1)
iris_stand_df.head()
```

Out[26]:

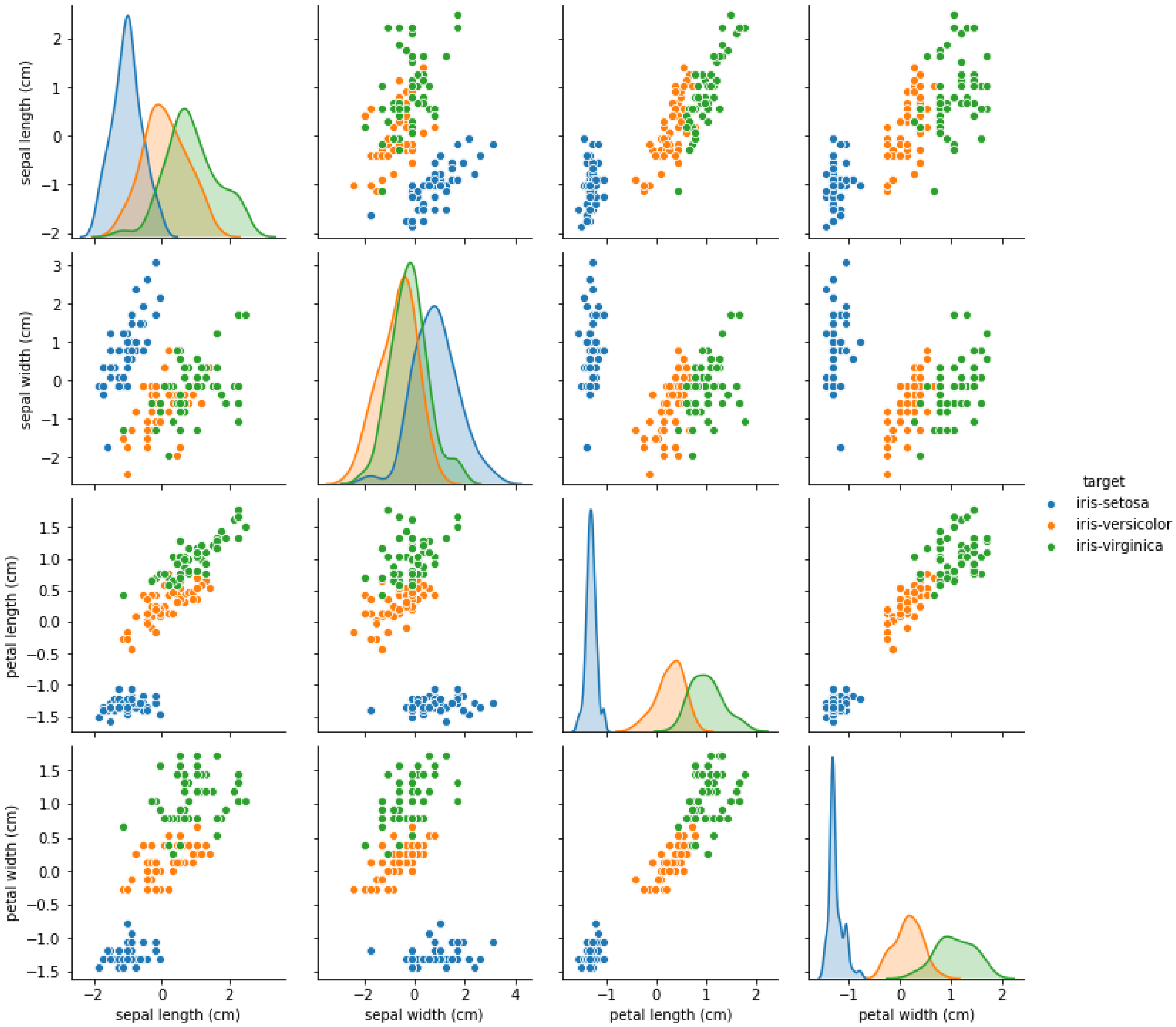
	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	target
0	-0.900681	1.019004	-1.340227	-1.315444	0
1	-1.143017	-0.131979	-1.340227	-1.315444	0
2	-1.385353	0.328414	-1.397064	-1.315444	0
3	-1.506521	0.098217	-1.283389	-1.315444	0
4	-1.021849	1.249201	-1.340227	-1.315444	0

In [37]:
```
class_label = {0:'iris-setosa',1:'iris-versicolor',2:'iris-virginica'}
```

In [38]:
```
iris_stand_df['target'] = iris_stand_df['target'].map(lambda row: class_label[row])
```

In [39]:
```
sns.pairplot(data=iris_stand_df,hue='target')
```

Out[39]: <seaborn.axisgrid.PairGrid at 0x20f8afe2438>



In [28]:
```
iris_stand_df.shape
```

Out[28]:
```
(150, 5)
```

In [29]:
```
iris_stand_df.target.value_counts()
```

Out[29]:
```
2    50
1    50
0    50
Name: target, dtype: int64
```

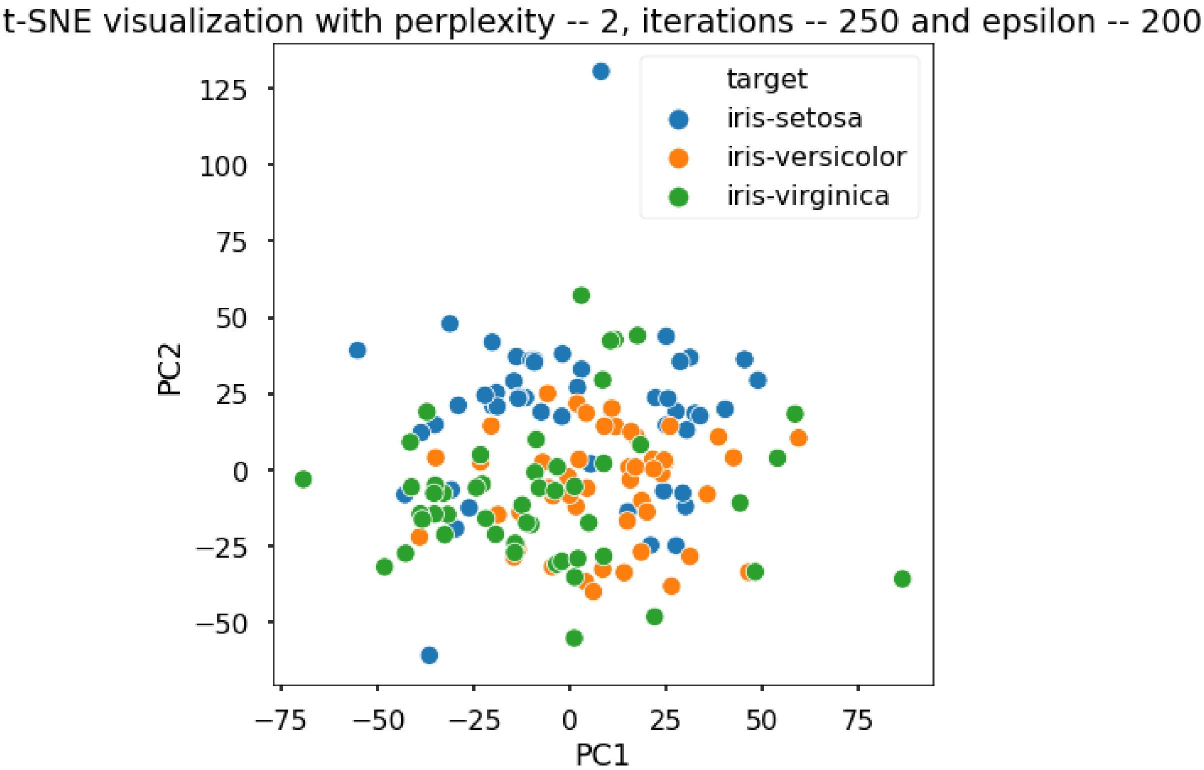
Multiple_runs

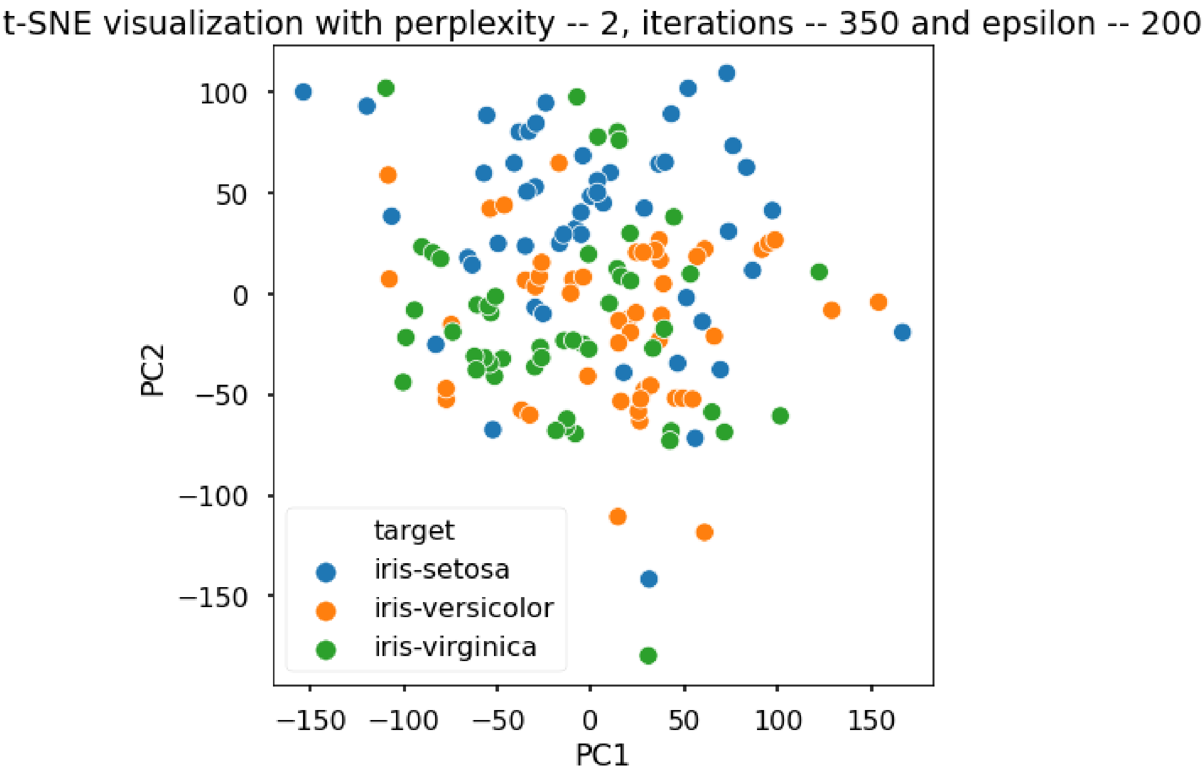
IRIS:CASE-I-Multiple_runs

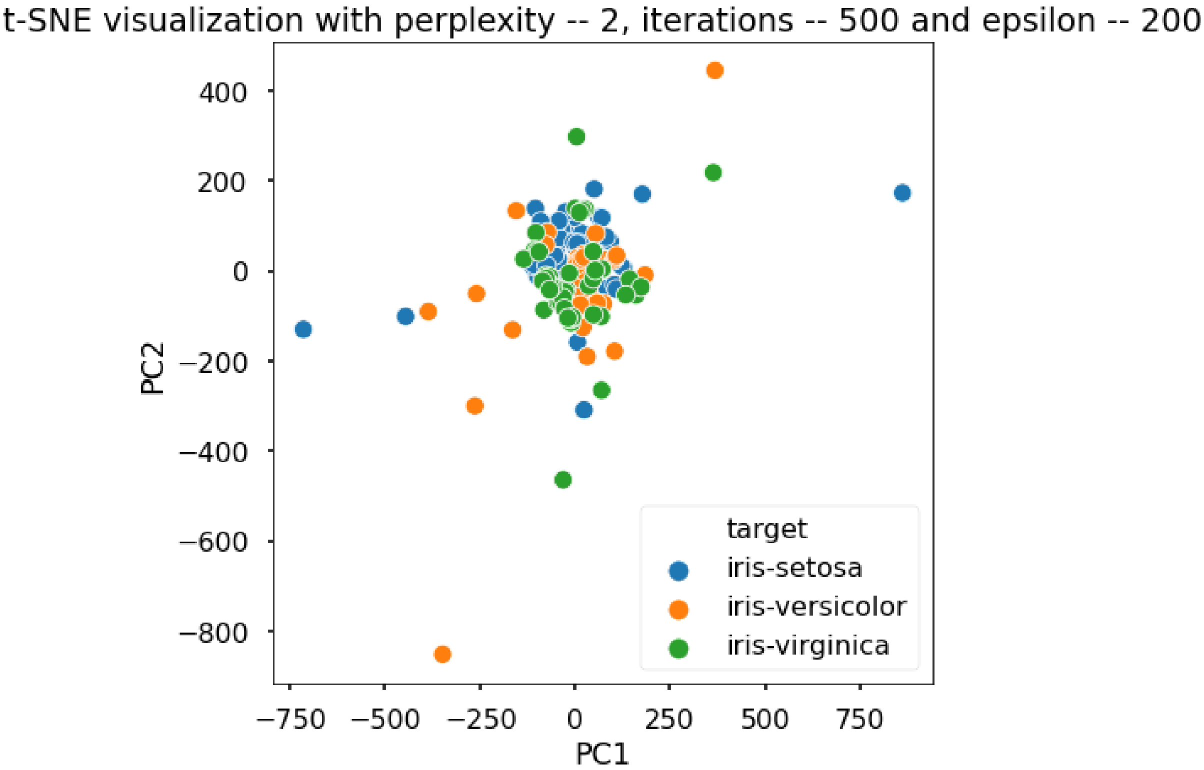
Running t-SNE on a range of perplexities and iterations with fixed learning rate and embedding initialization as 'random'

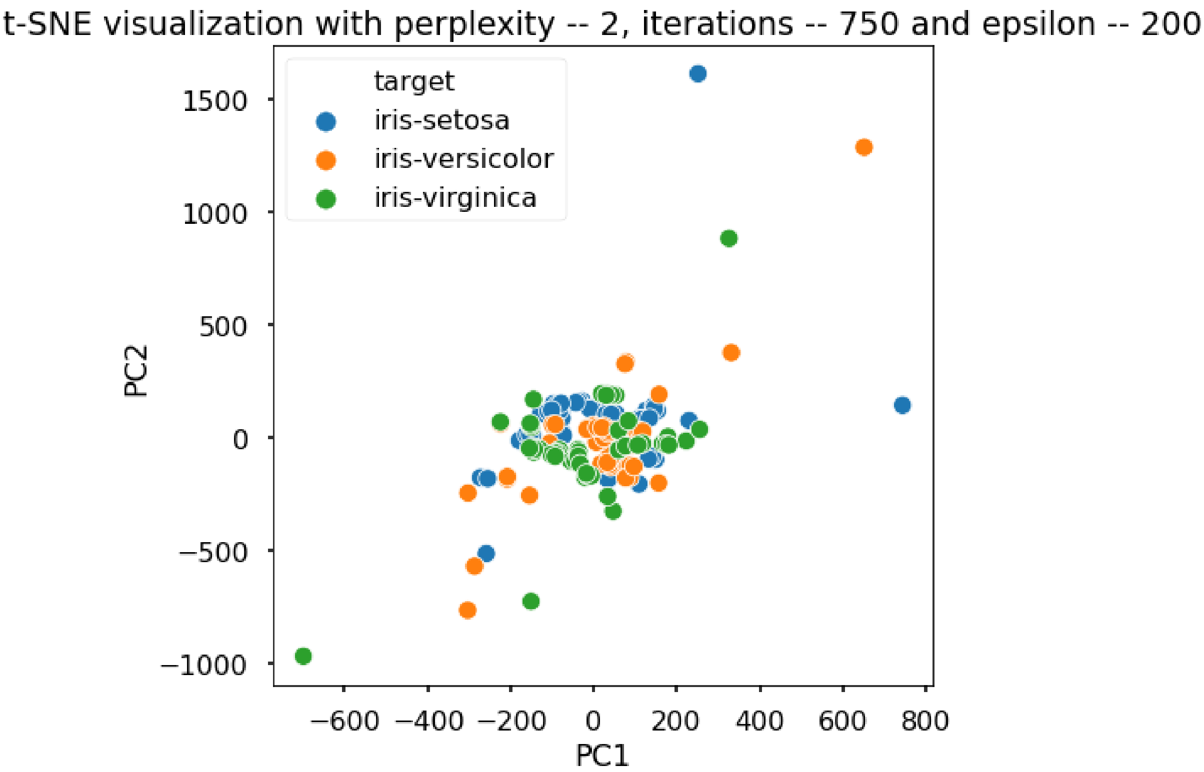
In [78]:
```
perplexities = [1,2,5,30,50,100,150]
iterations = [250,350,500,750,1000,2500,3500,4000,5000]

for pidx in range(len(perplexities)):
    for idx in range(len(iterations)):
        tsne_iris_1 = TSNE(n_components=2,perplexity=perplexities[pidx],learning_rat
```

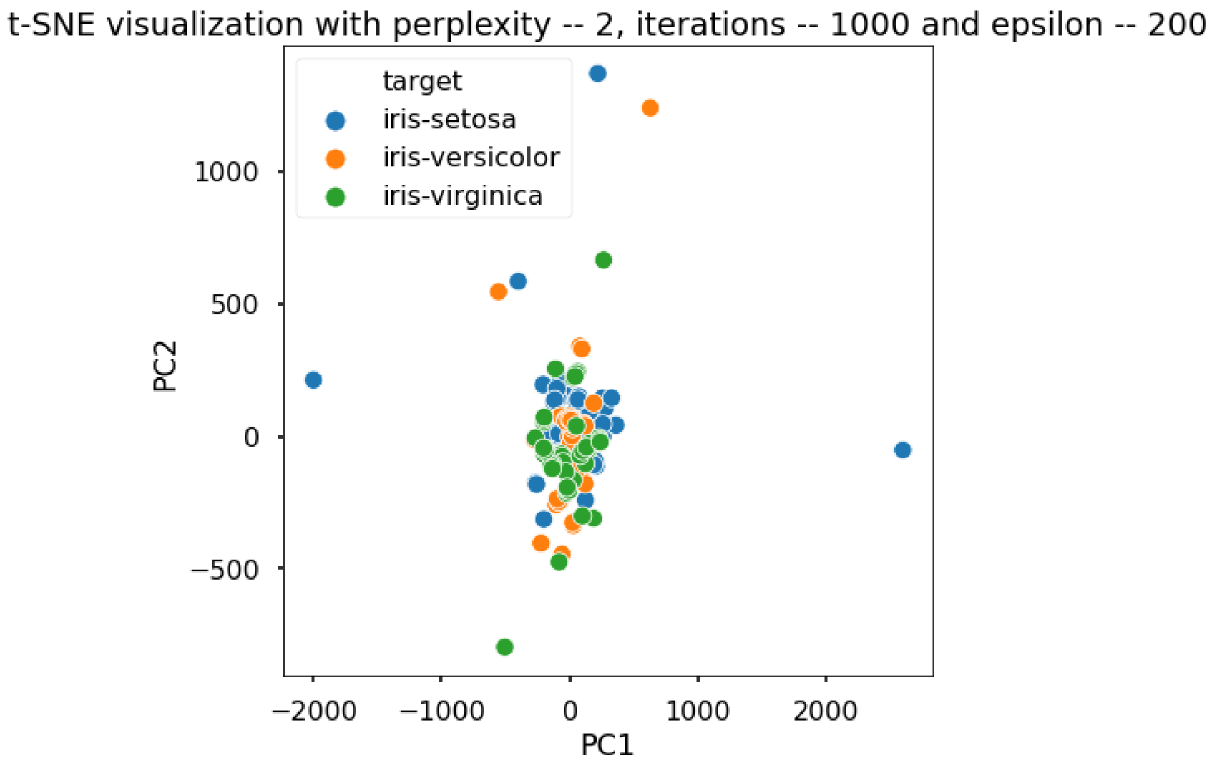
```
                                 init='random',n_jobs=-1,random_state=23)
        iris1_tsne_pcmps = pd.DataFrame(tsne_iris_1.fit_transform(iris_stand_df.iloc
        iris1_tsne_pcmps = pd.concat([iris1_tsne_pcmps,iris_stand_df['target']],axis
        with plt.style.context('seaborn-poster'):
            plt.figure(figsize=(7,7))
            sns.scatterplot(data=iris1_tsne_pcmps,x='PC1',y='PC2',hue='target')
            plt.title("t-SNE visualization with perplexity -- {0}, iterations -- {1}


            plt.show()
```

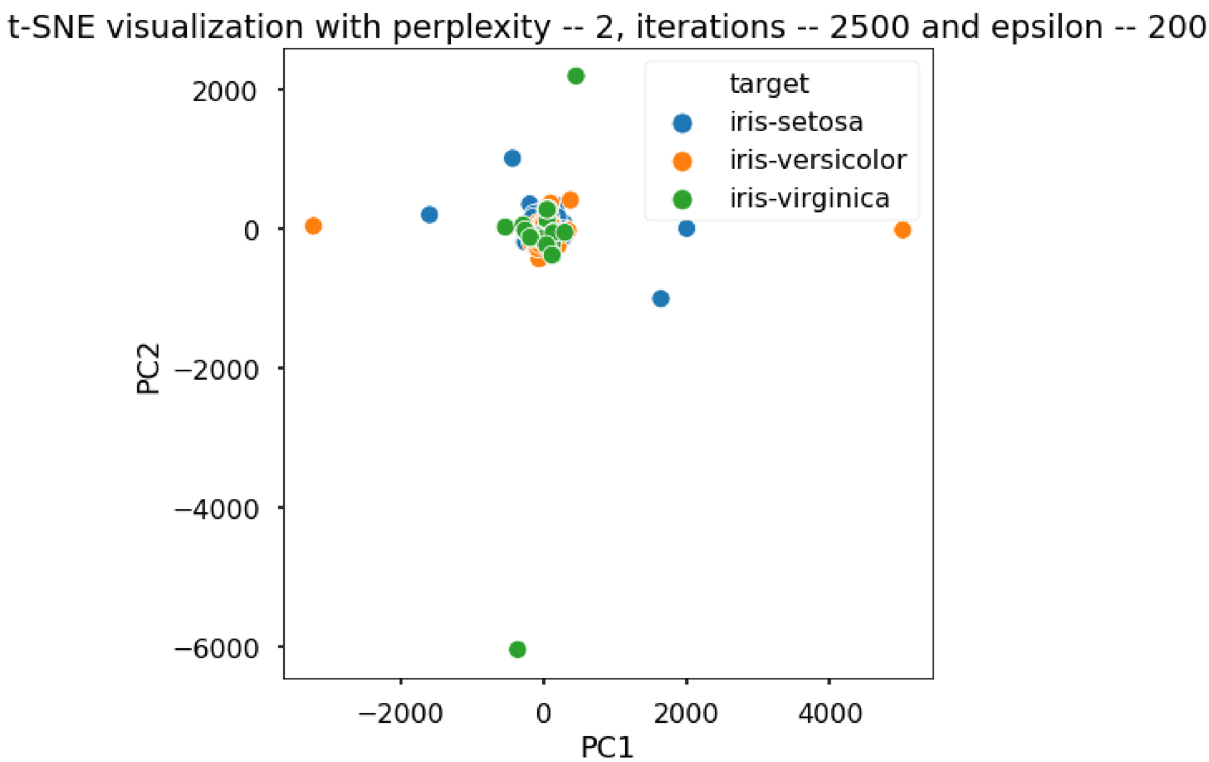


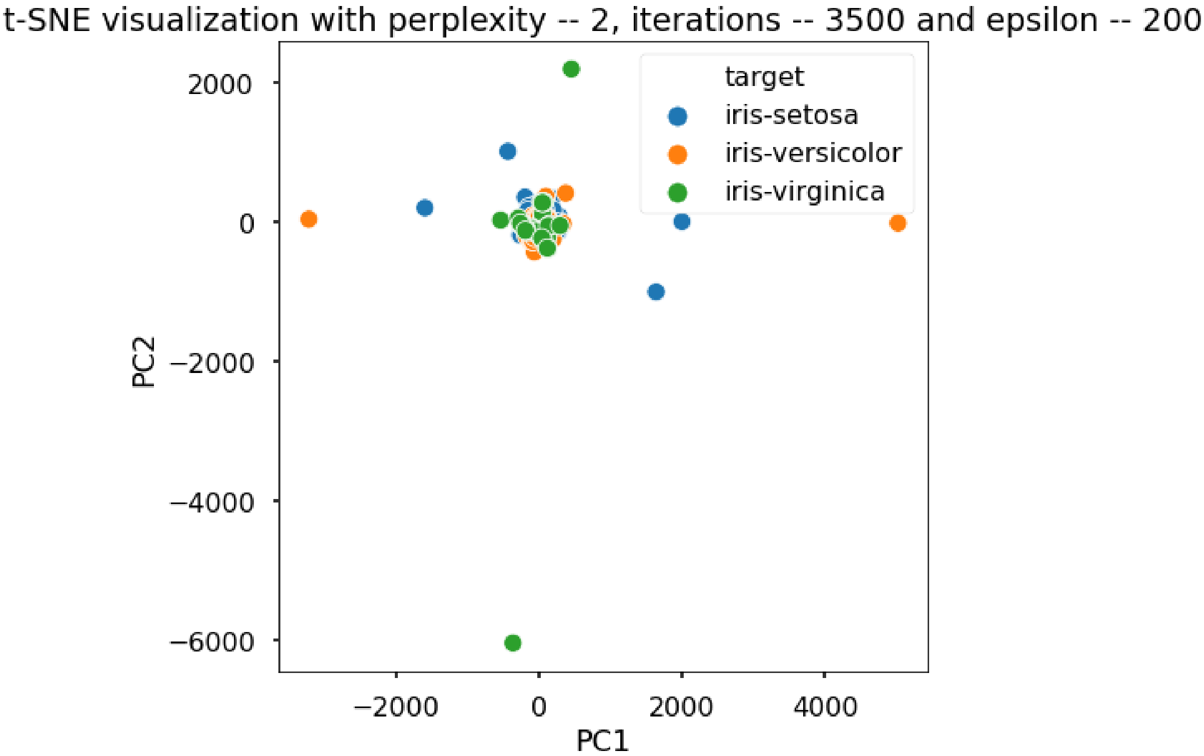


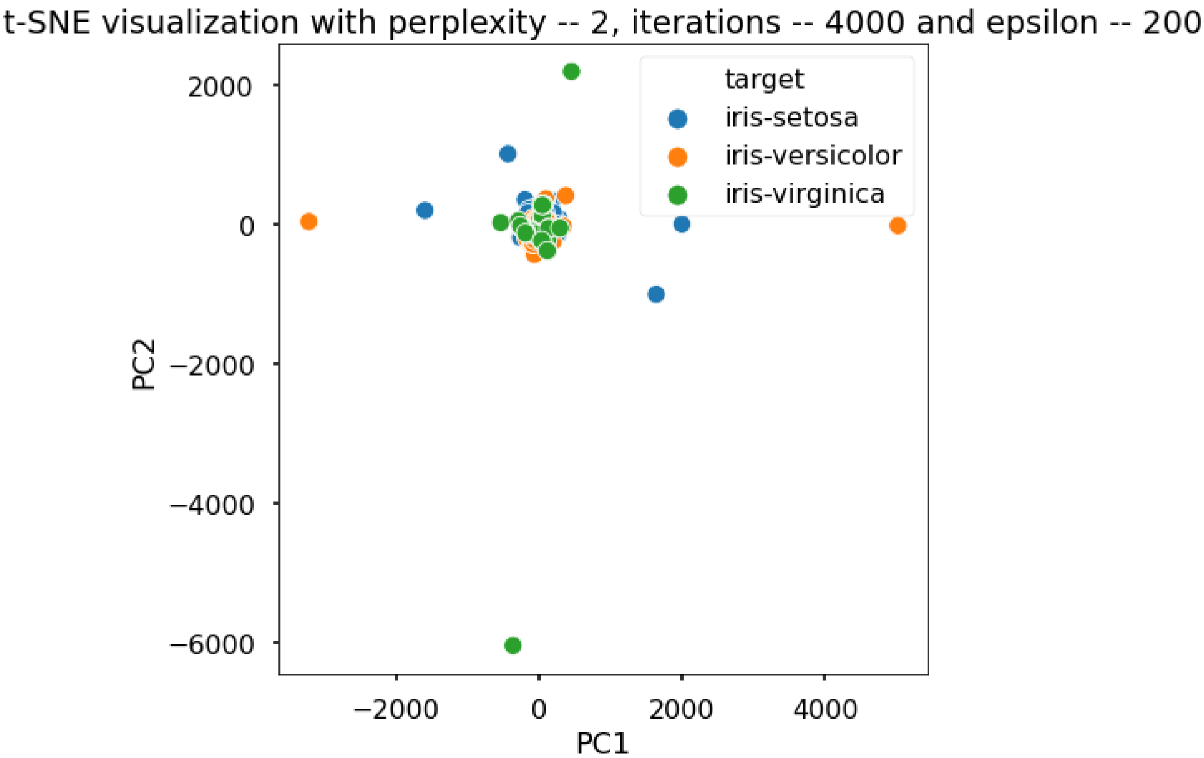

t-SNE visualization with perplexity -- 2, iterations -- 500 and epsilon -- 200
400
200
0
−200
−400
−600
−800
PC2
−750 −500 −250 0 250 500 750
PC1
target
iris-setosa
iris-versicolor
iris-virginica

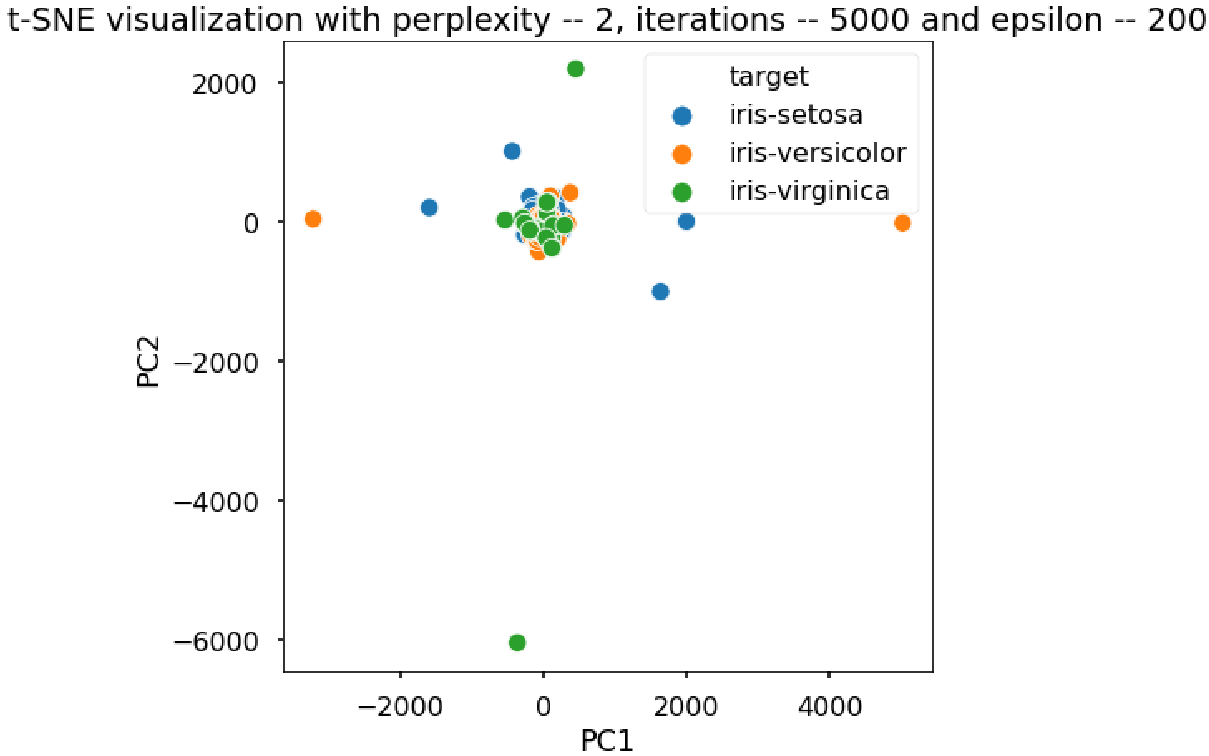

t-SNE visualization with perplexity -- 2, iterations -- 750 and epsilon -- 200
1500
1000
500
0
−500
−1000
PC2
−600 −400 −200 0 200 400 600 800
PC1
target
iris-setosa
iris-versicolor
iris-virginica


t-SNE visualization with perplexity -- 2, iterations -- 1000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
1000
500
0
−500
PC2
−2000
−1000
0
1000
2000
PC1

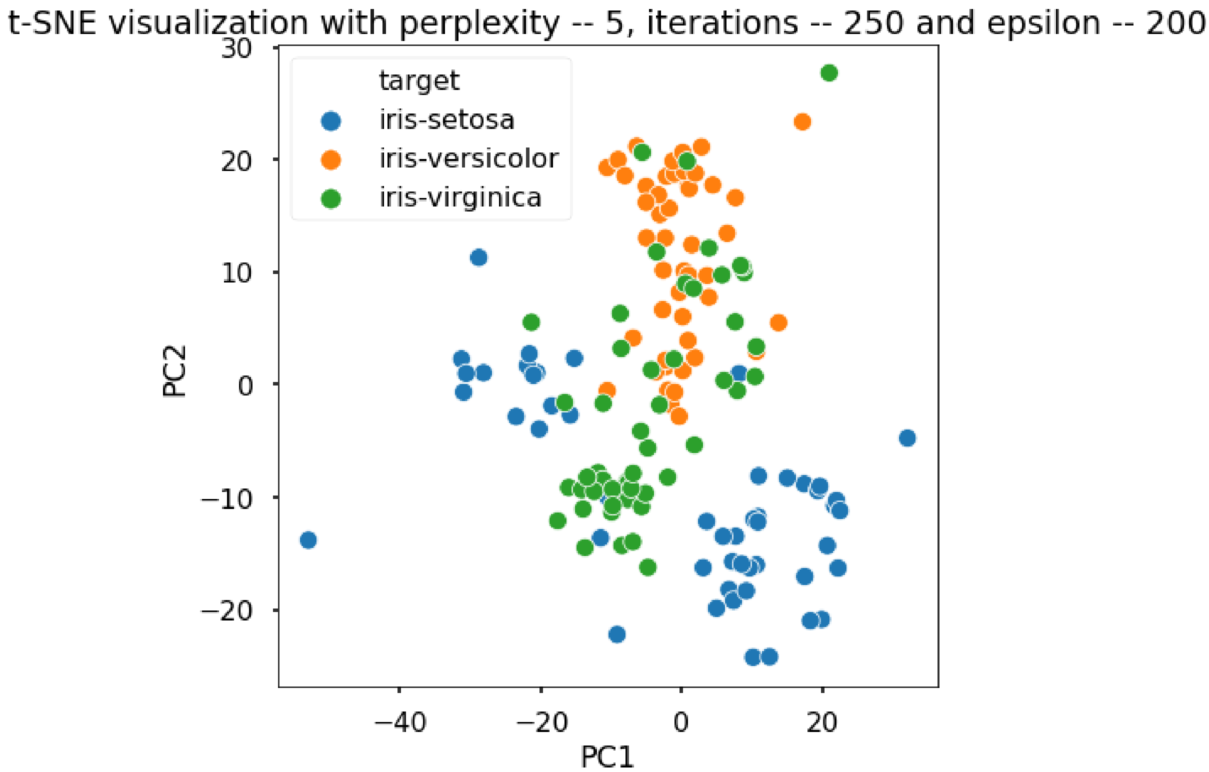

t-SNE visualization with perplexity -- 2, iterations -- 2500 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
2000
0
−2000
−4000
−6000
PC2
−2000
0
2000
4000
PC1


t-SNE visualization with perplexity -- 2, iterations -- 3500 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
2000
0
−2000
−4000
−6000
−2000
0
2000
4000

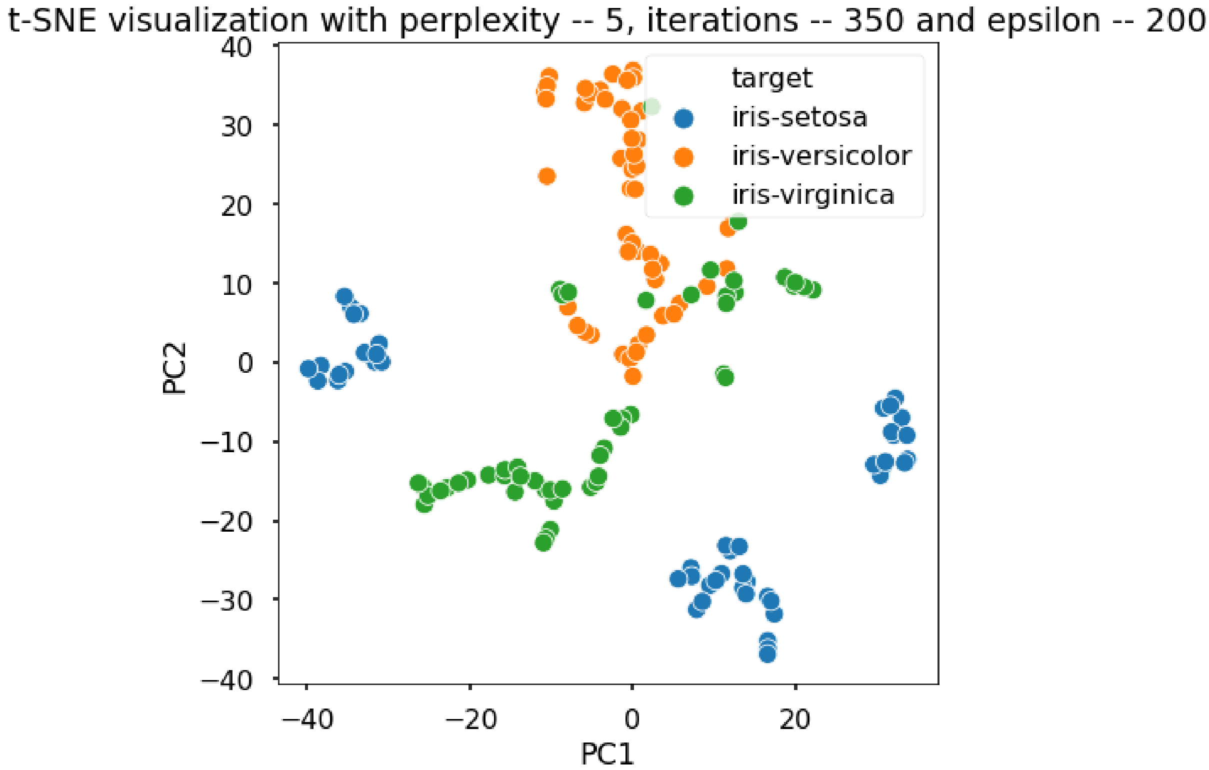

t-SNE visualization with perplexity -- 2, iterations -- 4000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
2000
0
−2000
−4000
−6000
−2000
0
2000
4000

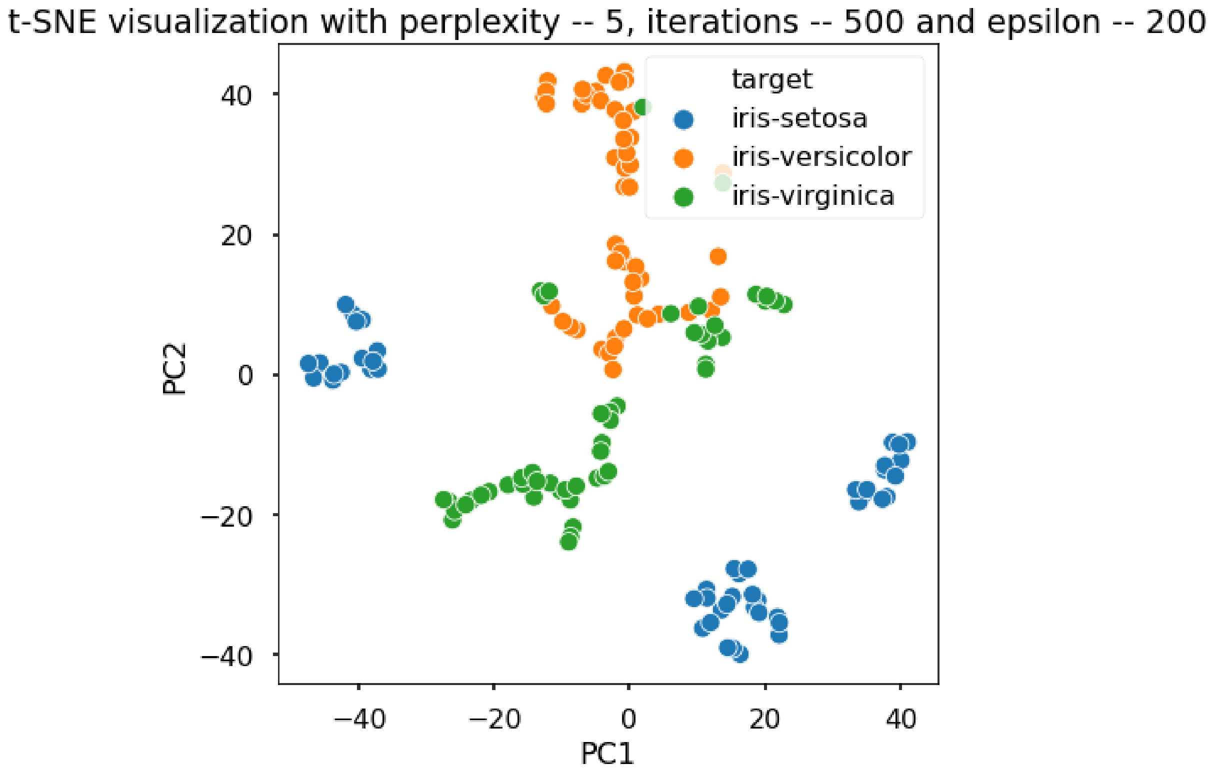


t-SNE visualization with perplexity -- 2, iterations -- 5000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
2000
0
−2000
−4000
−6000
PC2
−2000
0
2000
4000
PC1

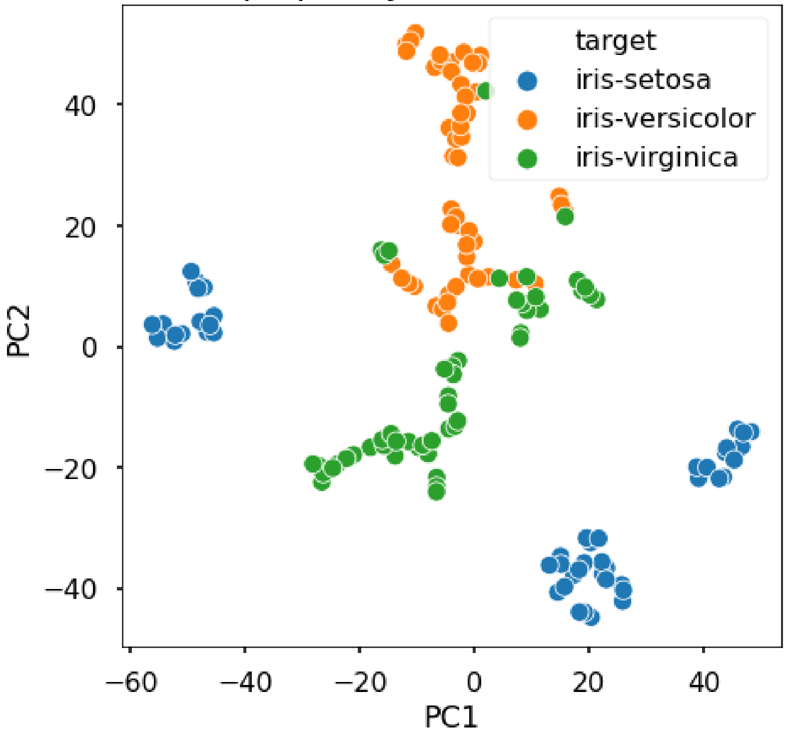


t-SNE visualization with perplexity -- 5, iterations -- 250 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
30
20
10
0
−10
−20
PC2
−40
−20
0
20
PC1

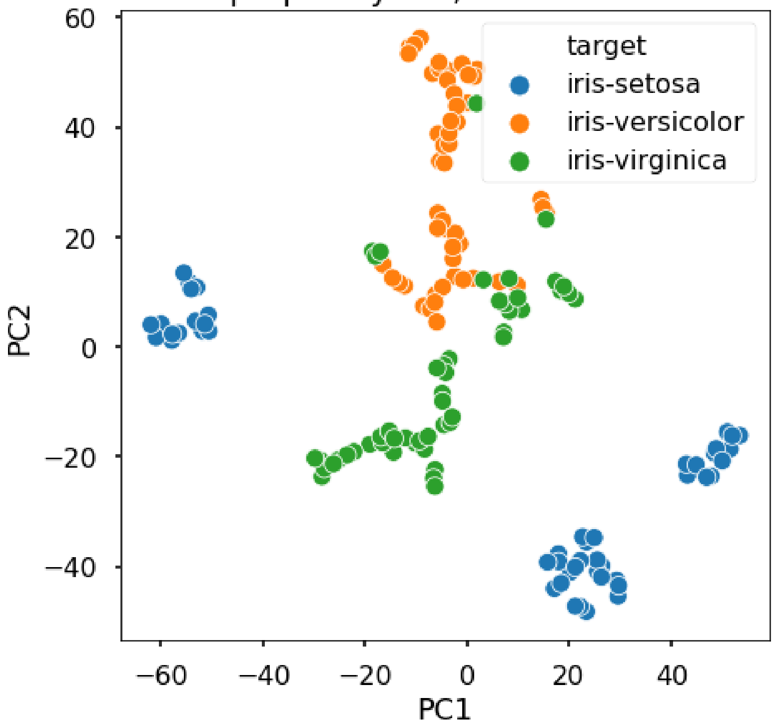

t-SNE visualization with perplexity -- 5, iterations -- 350 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC1
PC2
40
30
20
10
0
−10
−20
−30
−40
−40
−20
0
20


t-SNE visualization with perplexity -- 5, iterations -- 500 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC1
PC2
40
20
0
−20
−40
−40
−20
0
20
40

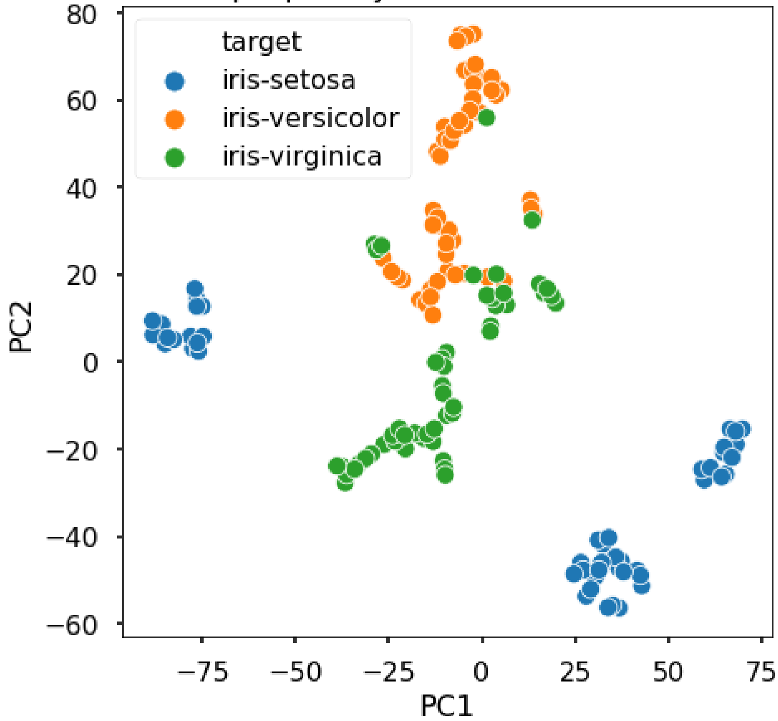
t-SNE visualization with perplexity -- 5, iterations -- 750 and epsilon -- 200



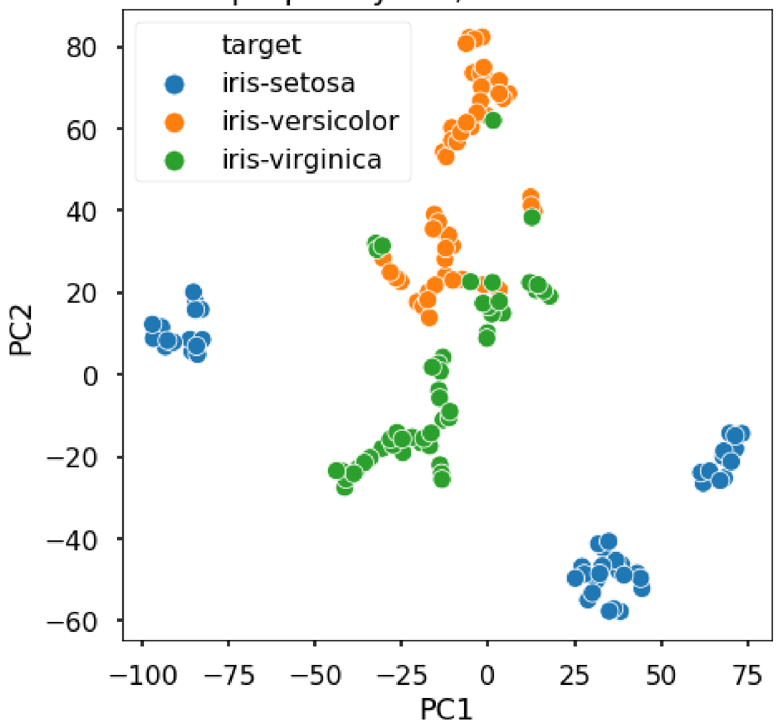
t-SNE visualization with perplexity -- 5, iterations -- 1000 and epsilon -- 200

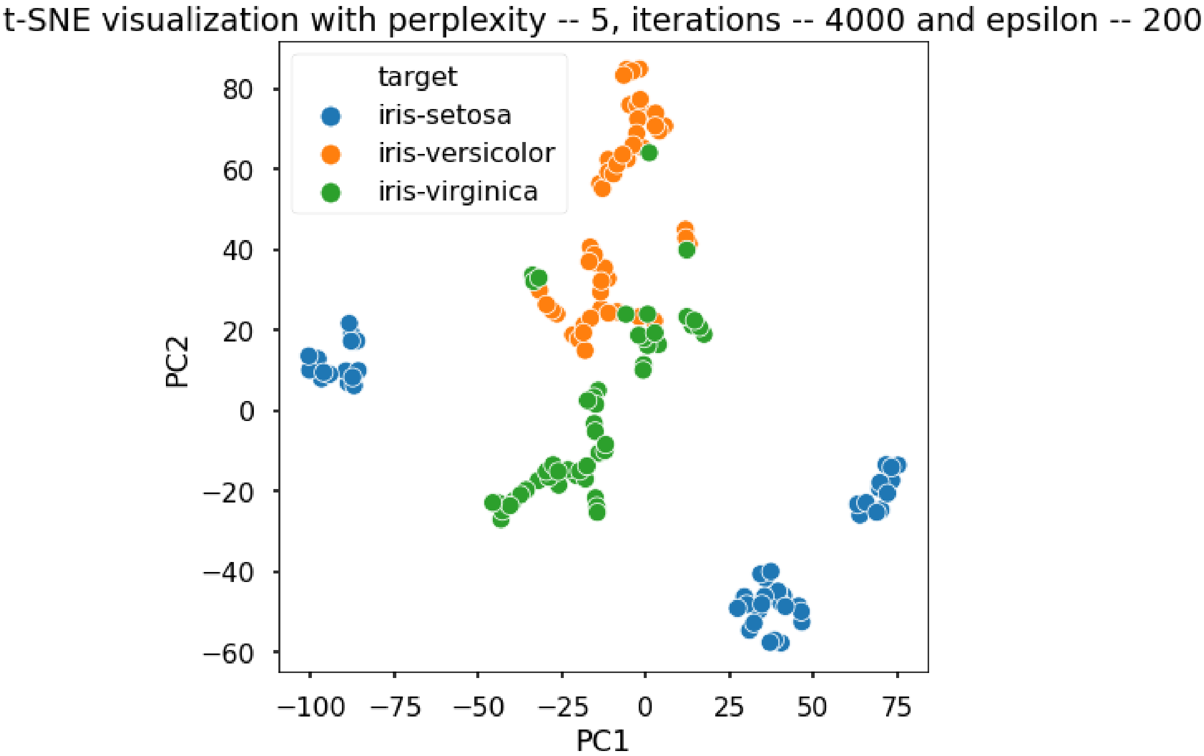


t-SNE visualization with perplexity -- 5, iterations -- 2500 and epsilon -- 200

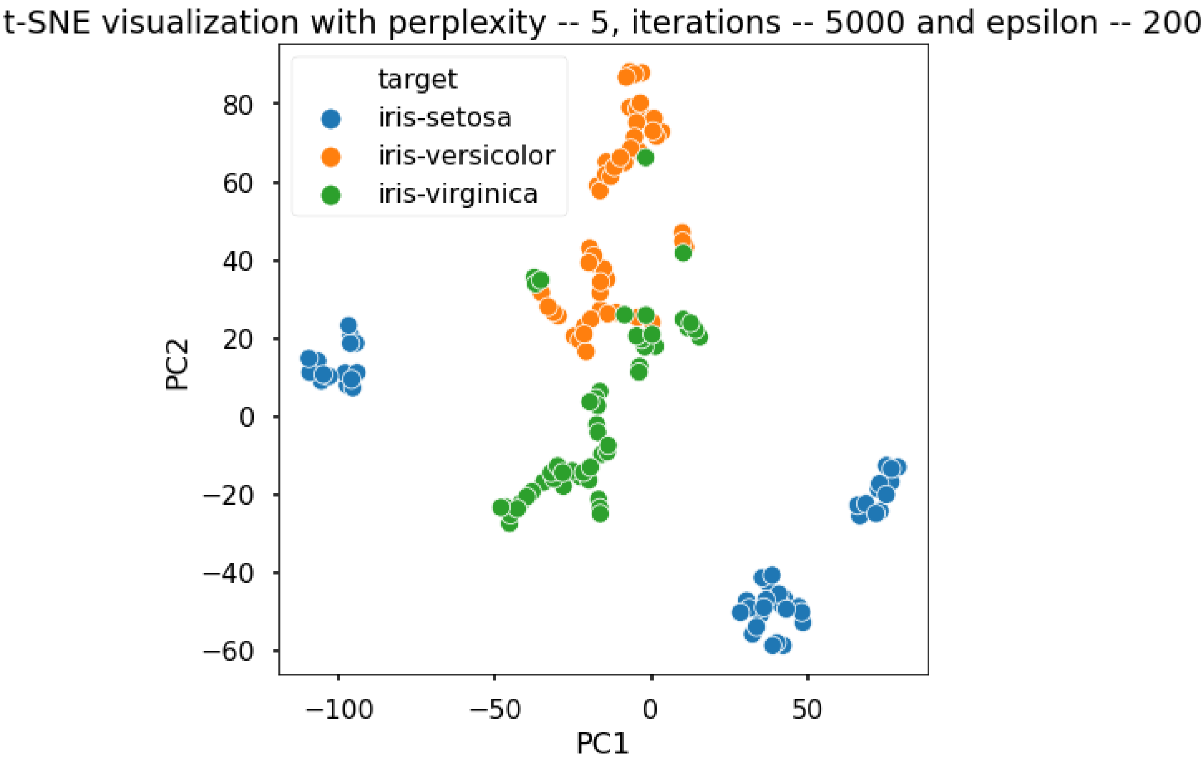


t-SNE visualization with perplexity -- 5, iterations -- 3500 and epsilon -- 200

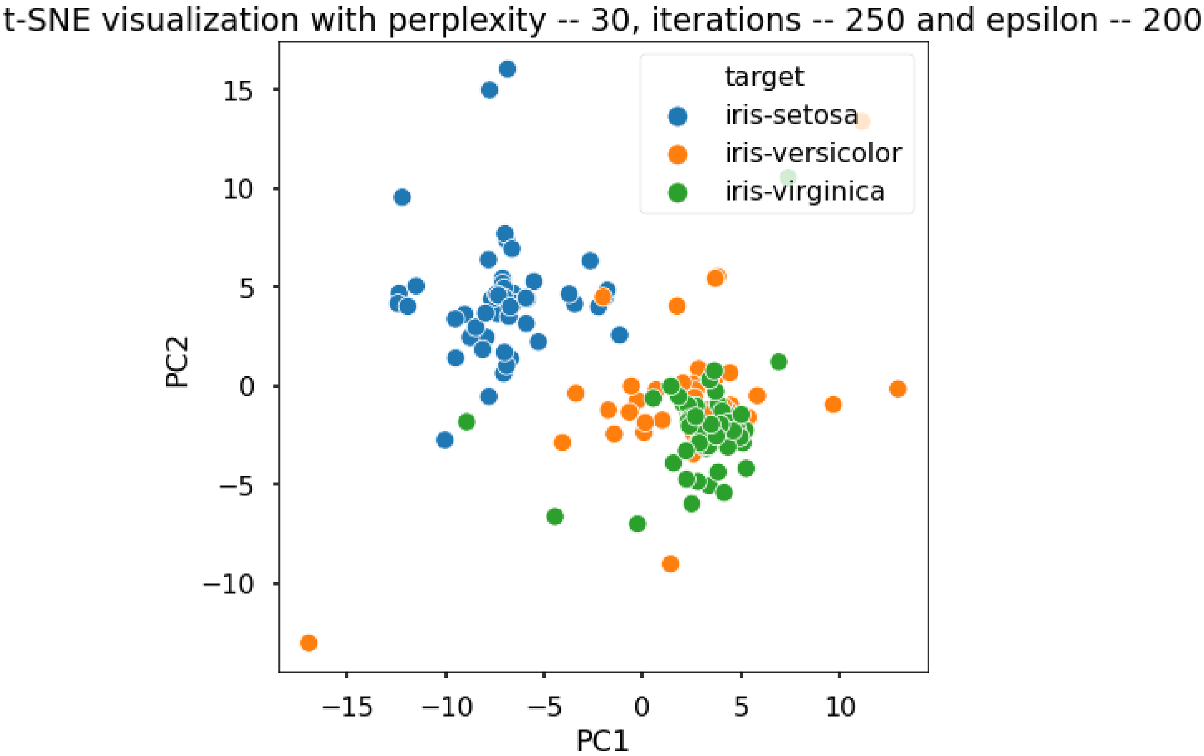


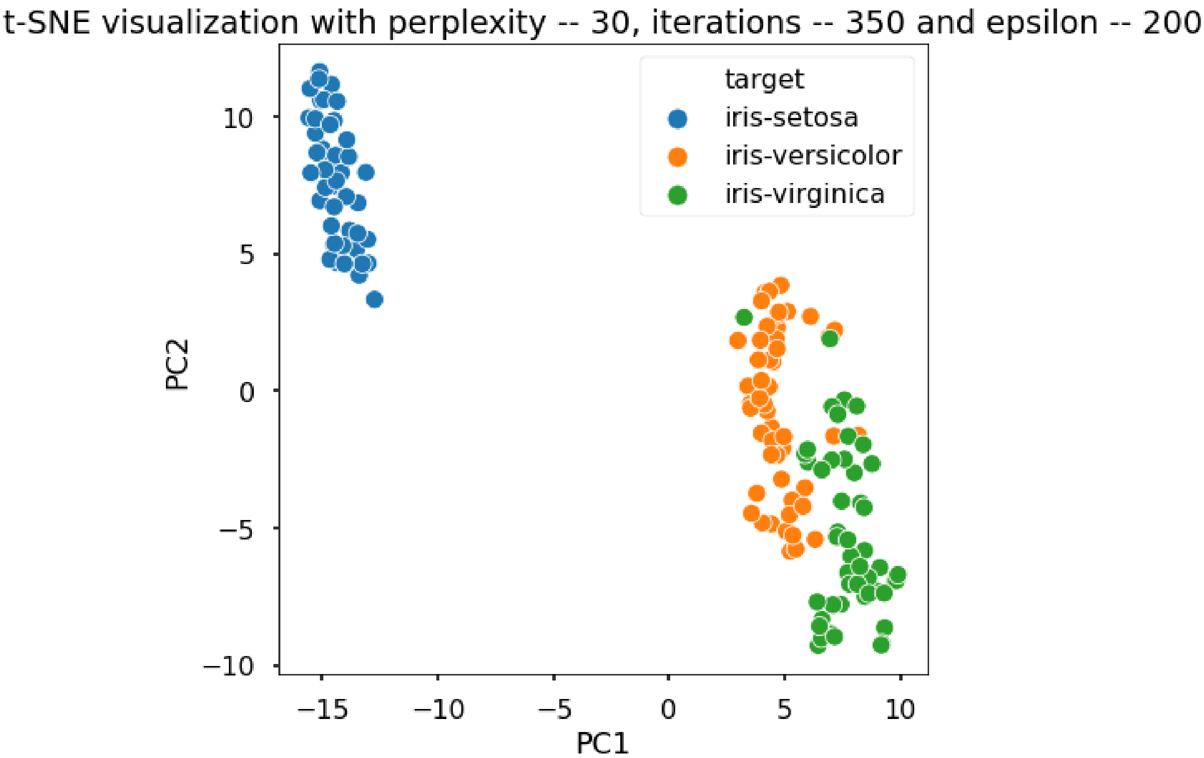


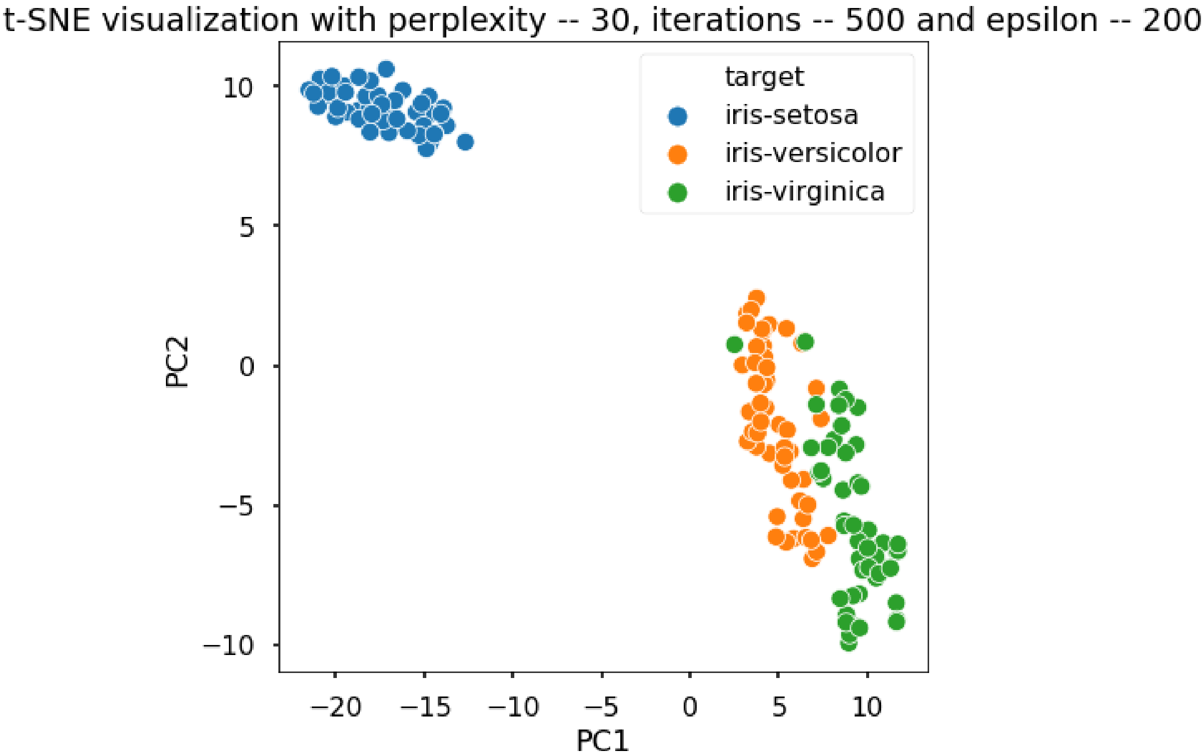
t-SNE visualization with perplexity -- 5, iterations -- 4000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
80
60
40
20
0
−20
−40
−60
−100
−75
−50
−25
0
25
50
75

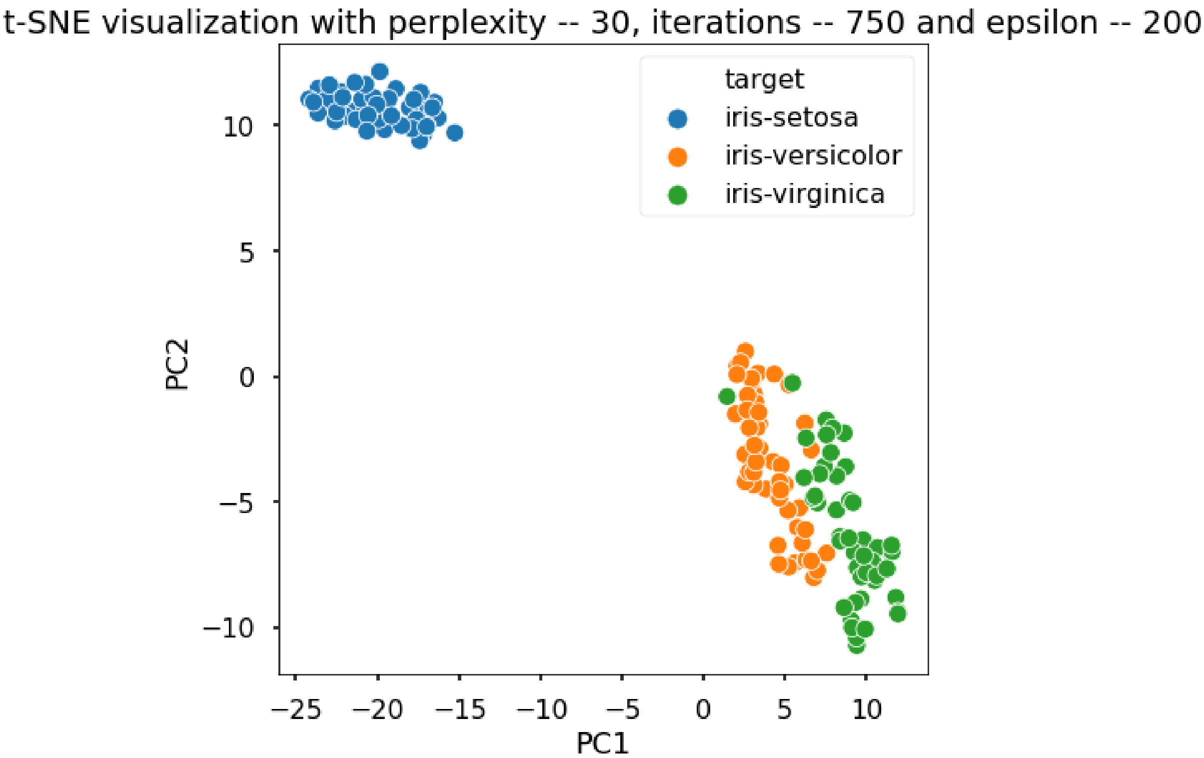


t-SNE visualization with perplexity -- 5, iterations -- 5000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
80
60
40
20
0
−20
−40
−60
−100
−50
0
50

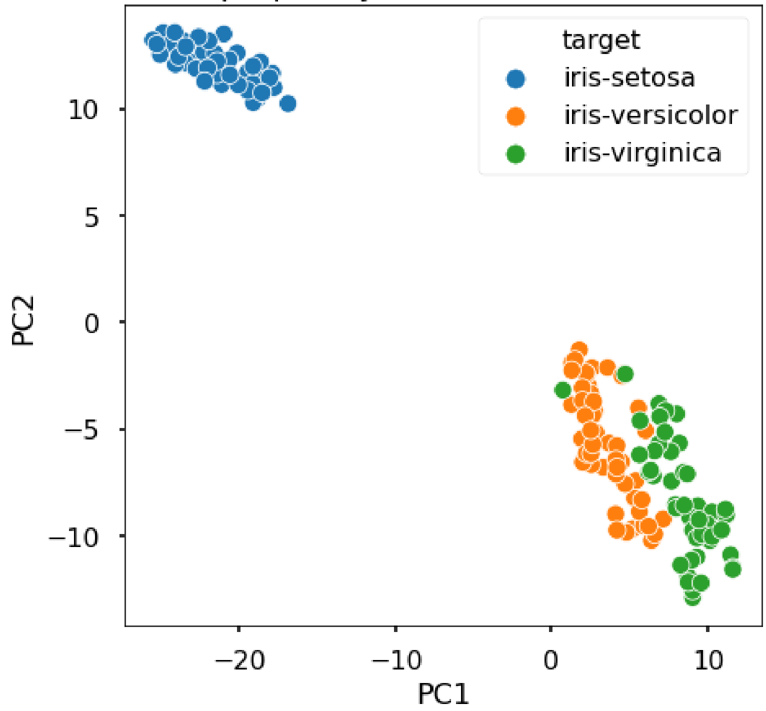

t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC1
PC2
15
10
5
0
−5
−10
−15
−10
−5
0
5
10


t-SNE visualization with perplexity -- 30, iterations -- 350 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC1
PC2
10
5
0
−5
−10
−15
−10
−5
0
5
10

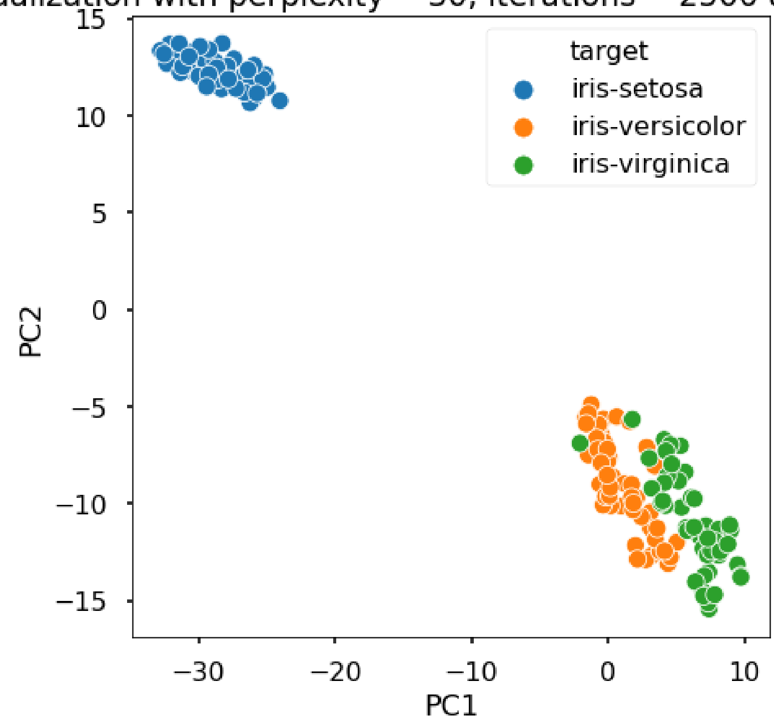

t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
10
5
0
−5
−10
−20
−15
−10
−5
0
5
10

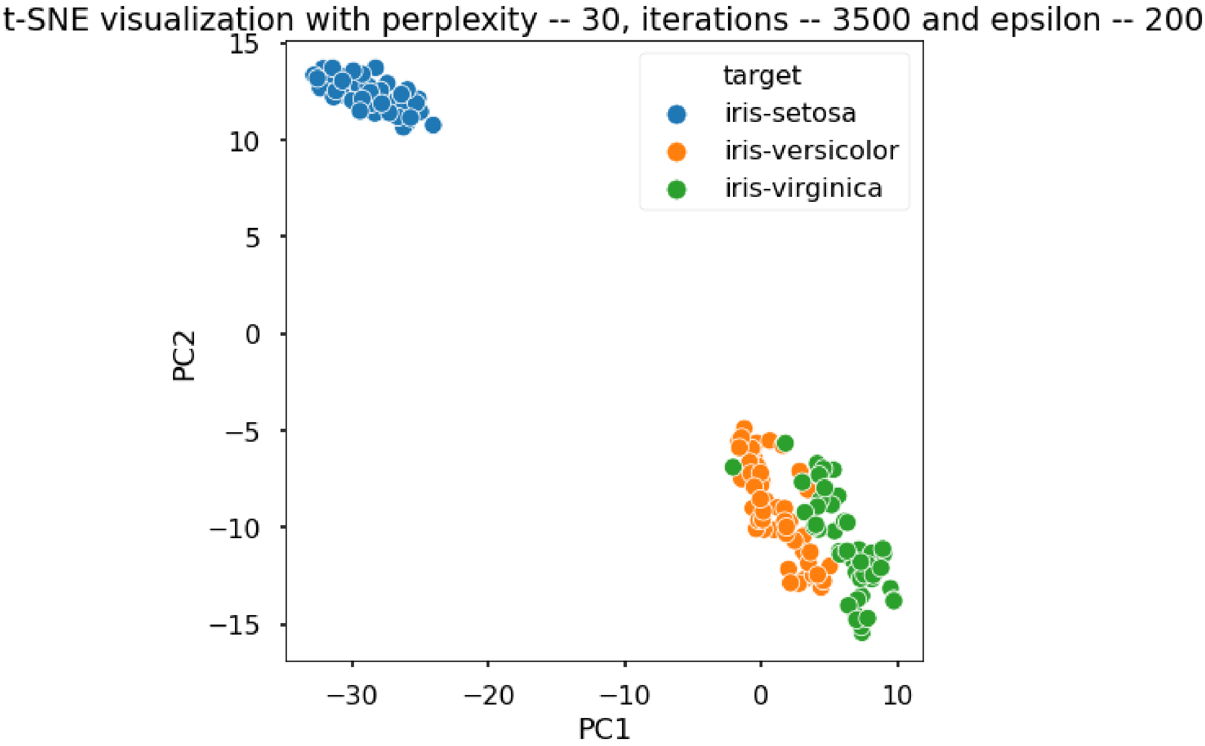

t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
10
5
0
−5
−10
−25
−20
−15
−10
−5
0
5
10

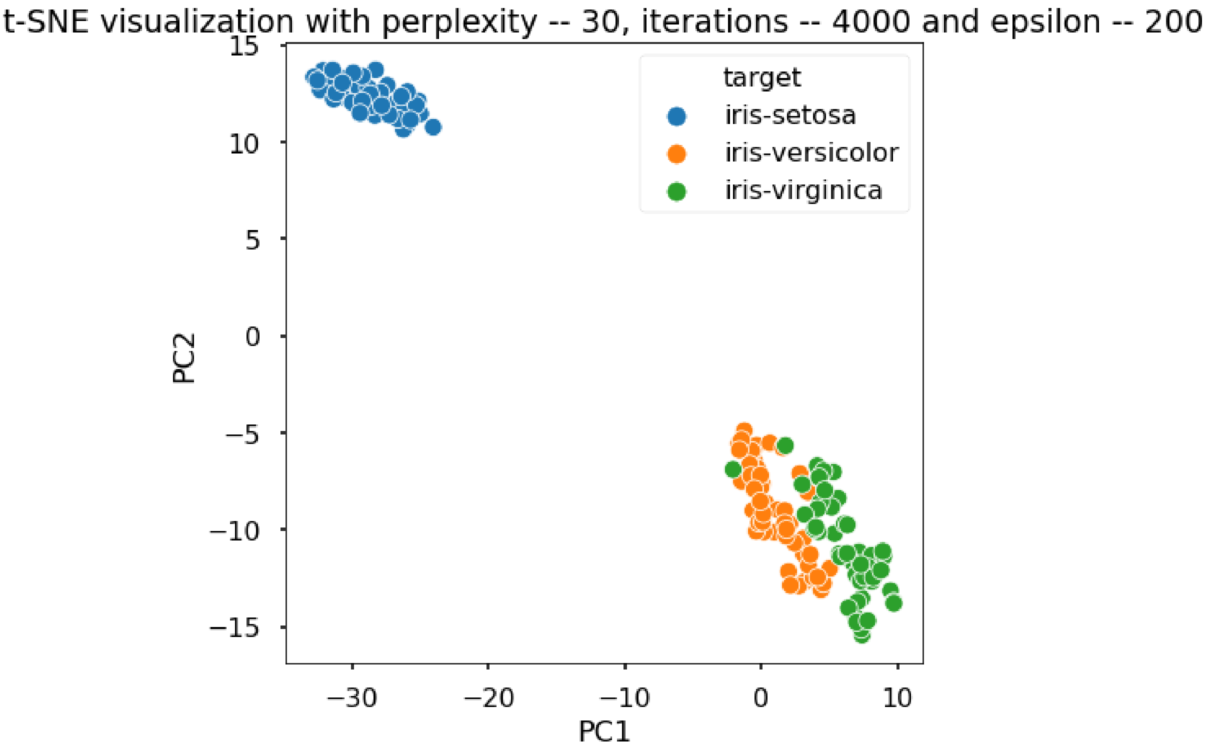
t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 200



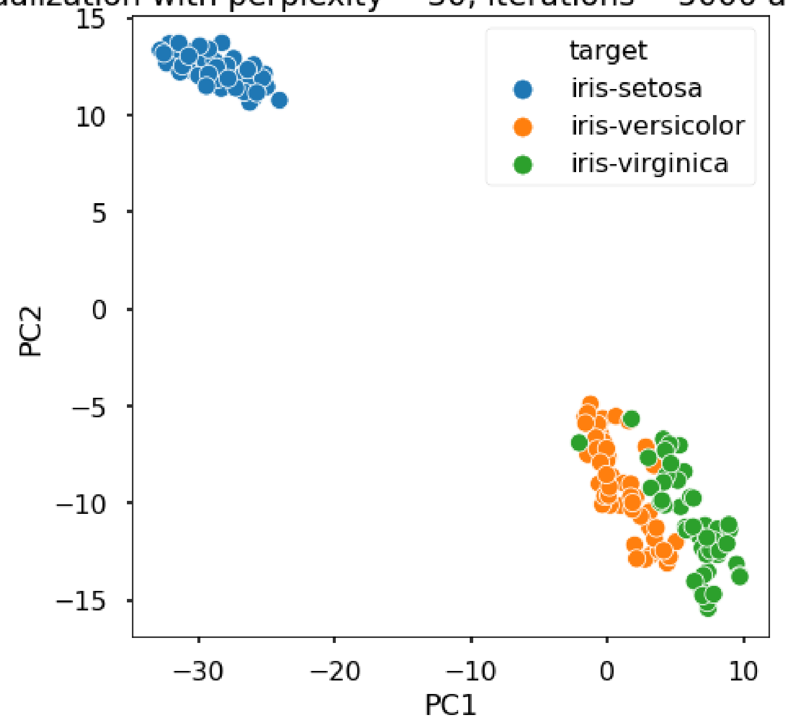
t-SNE visualization with perplexity -- 30, iterations -- 2500 and epsilon -- 200



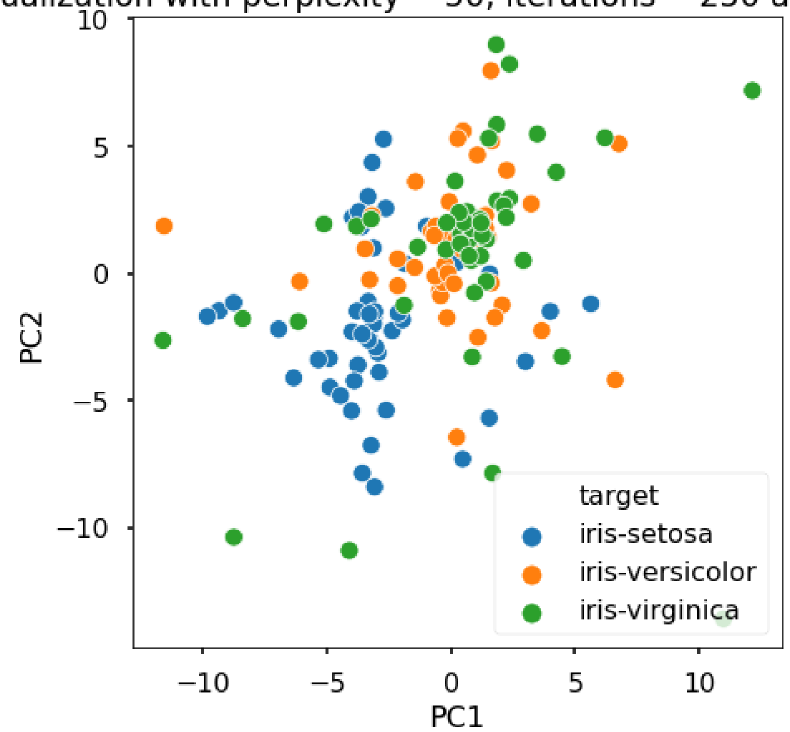

t-SNE visualization with perplexity -- 30, iterations -- 3500 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10


t-SNE visualization with perplexity -- 30, iterations -- 4000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10

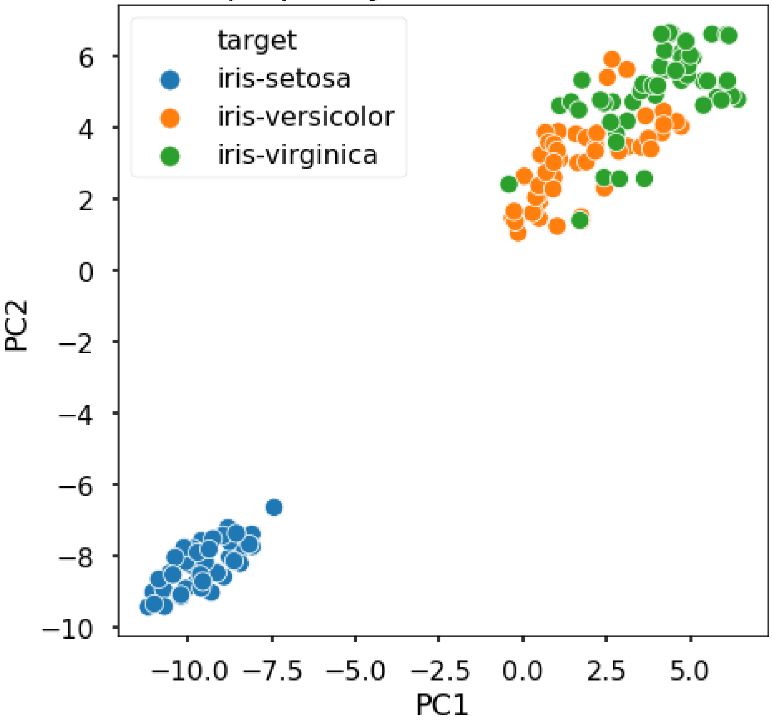
t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 200



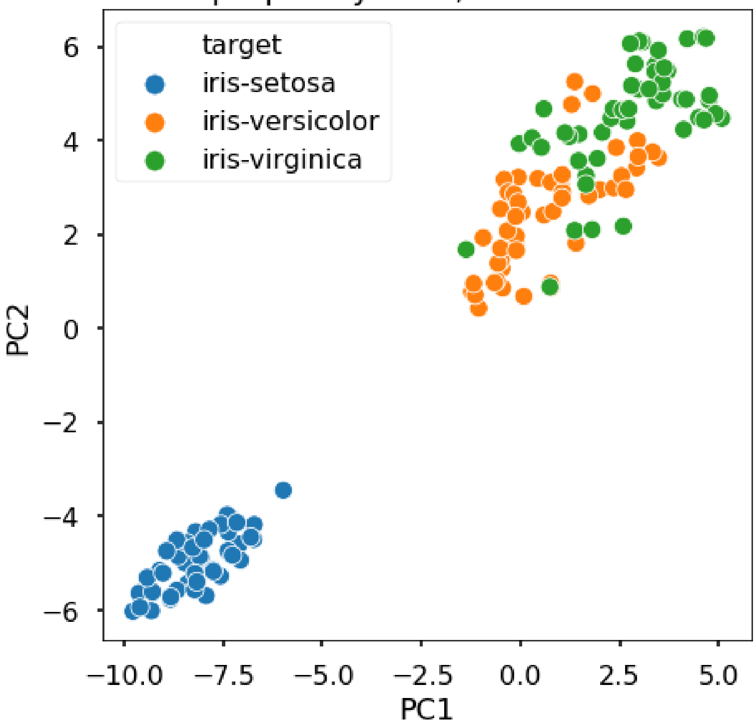
t-SNE visualization with perplexity -- 50, iterations -- 250 and epsilon -- 200

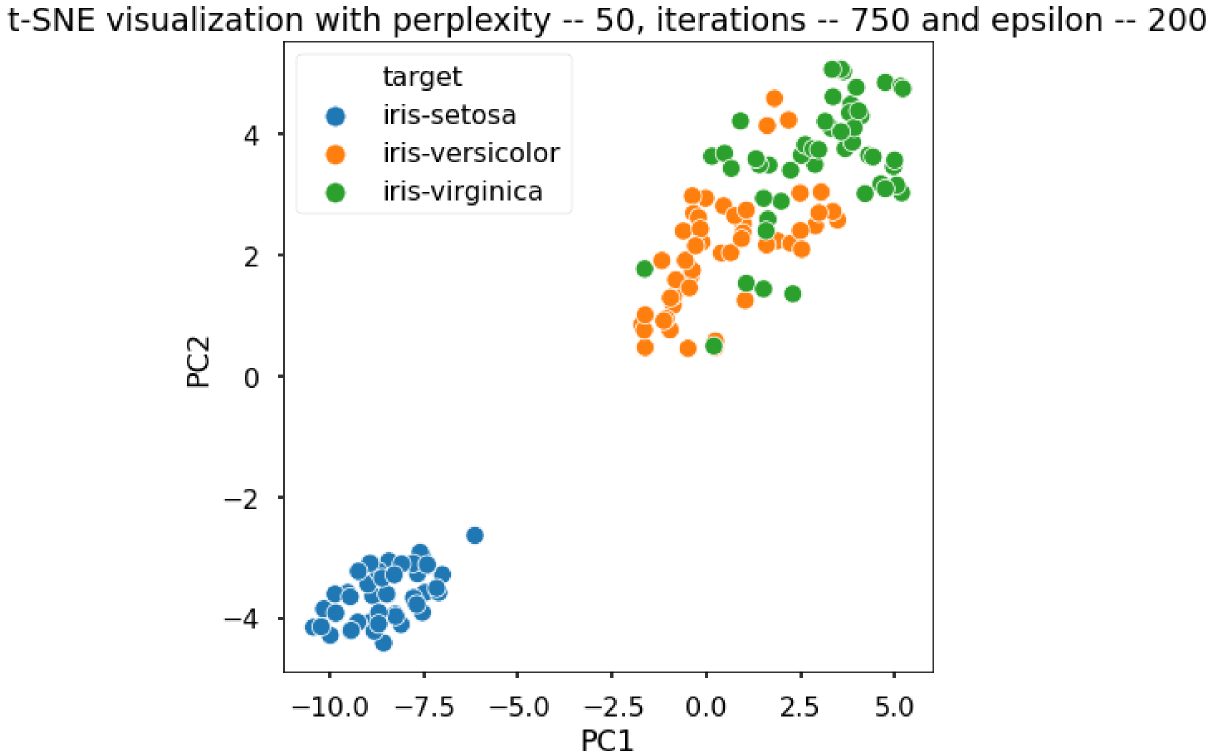


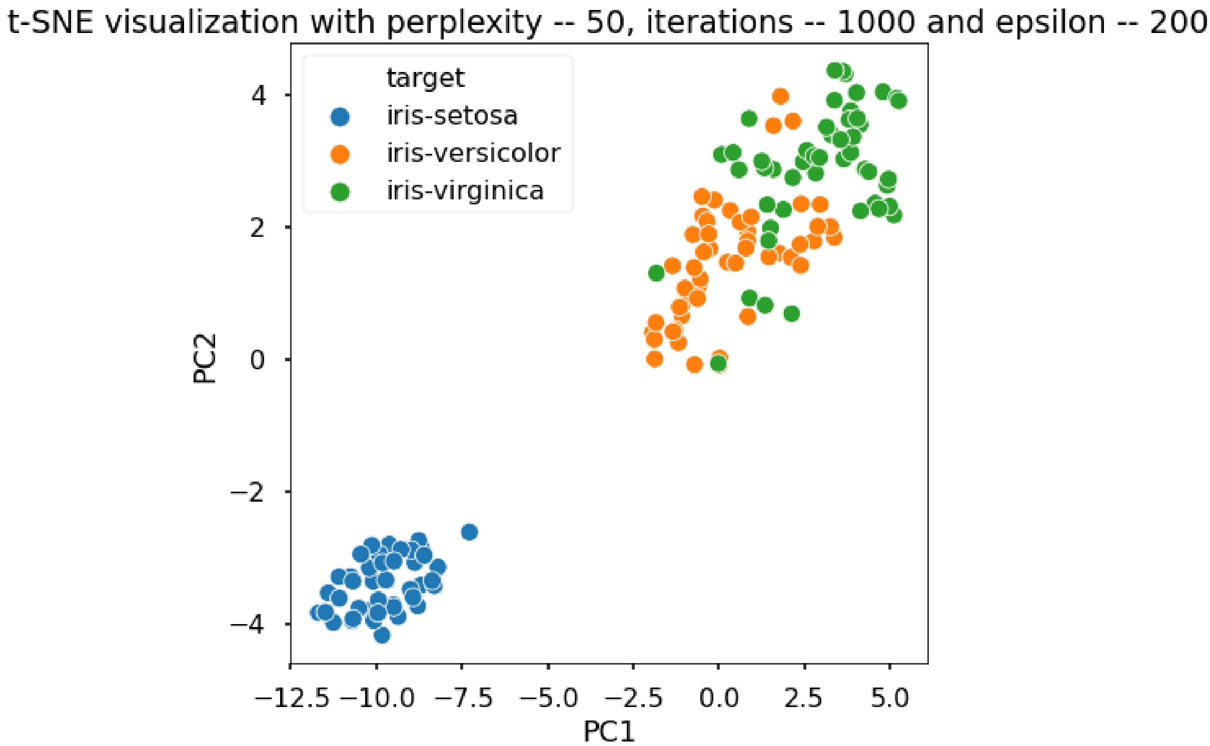
t-SNE visualization with perplexity -- 50, iterations -- 350 and epsilon -- 200



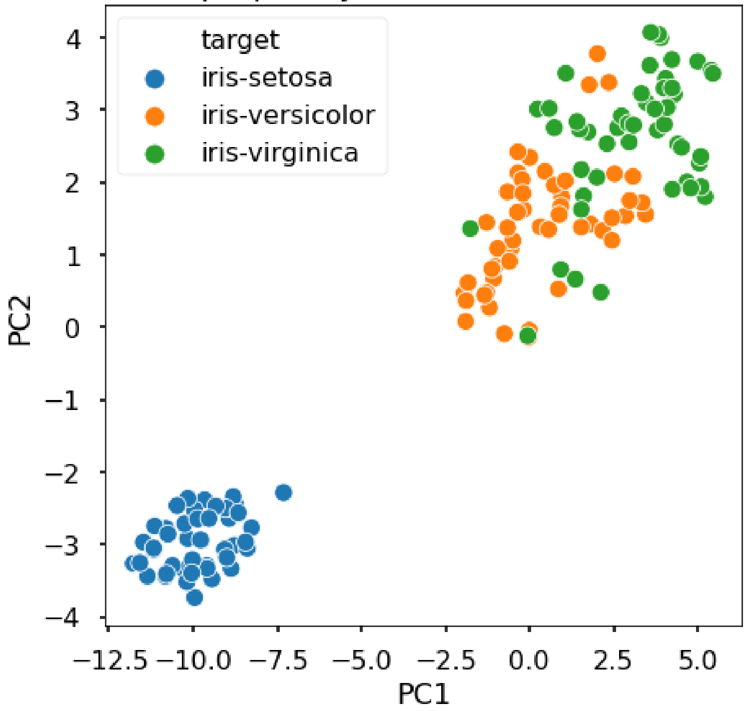
t-SNE visualization with perplexity -- 50, iterations -- 500 and epsilon -- 200



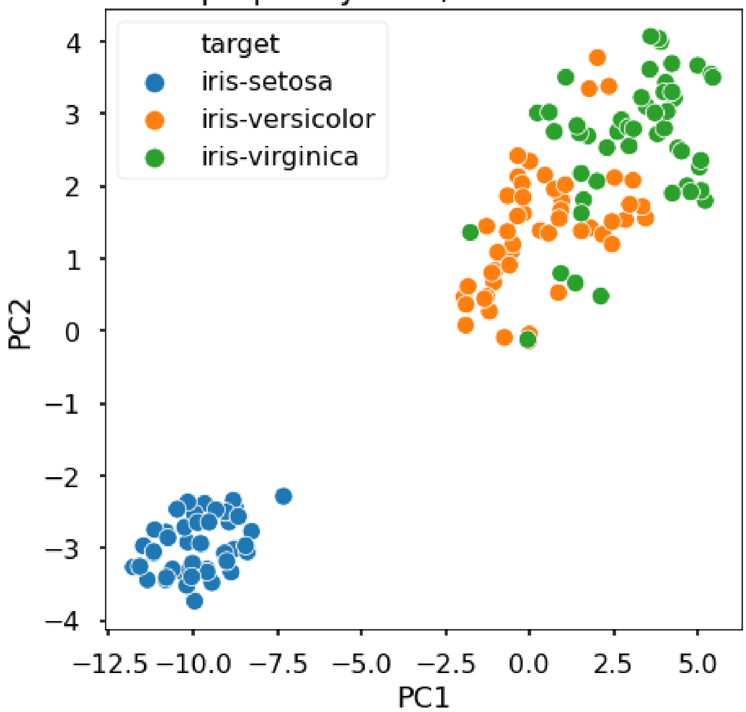

t-SNE visualization with perplexity -- 50, iterations -- 750 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
4
2
0
−2
−4
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0

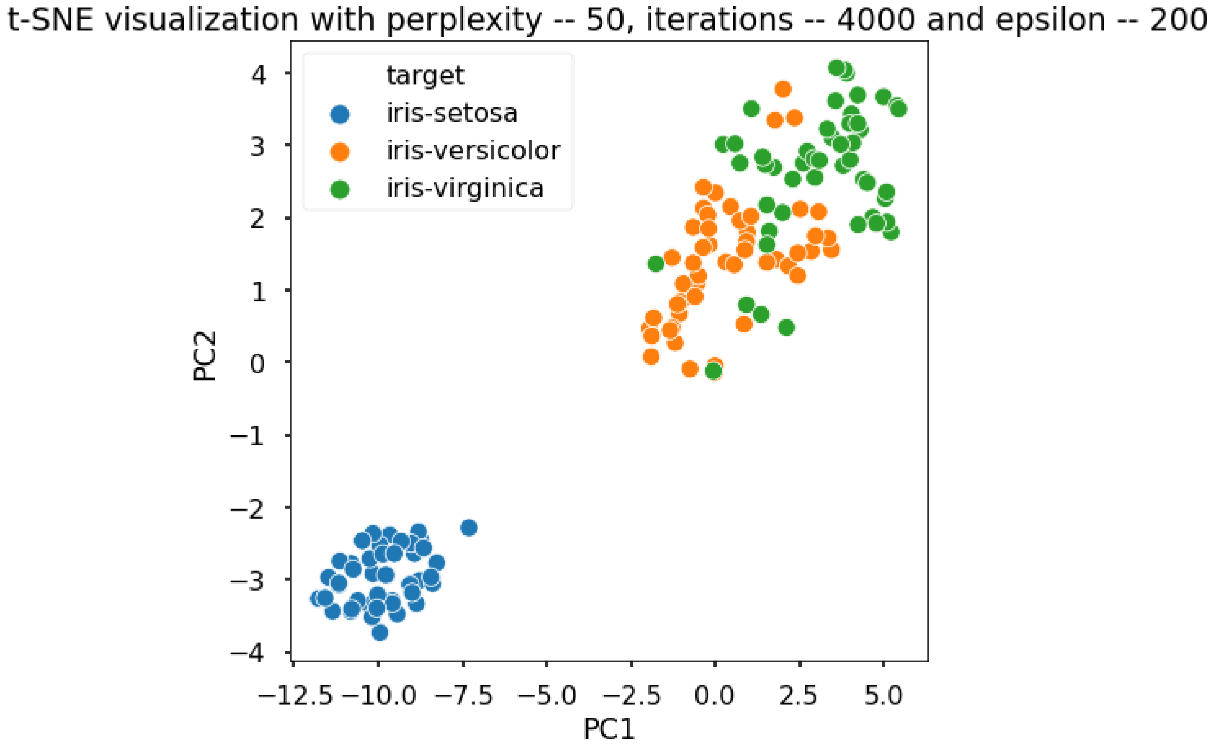

t-SNE visualization with perplexity -- 50, iterations -- 1000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
4
2
0
−2
−4
−12.5
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0

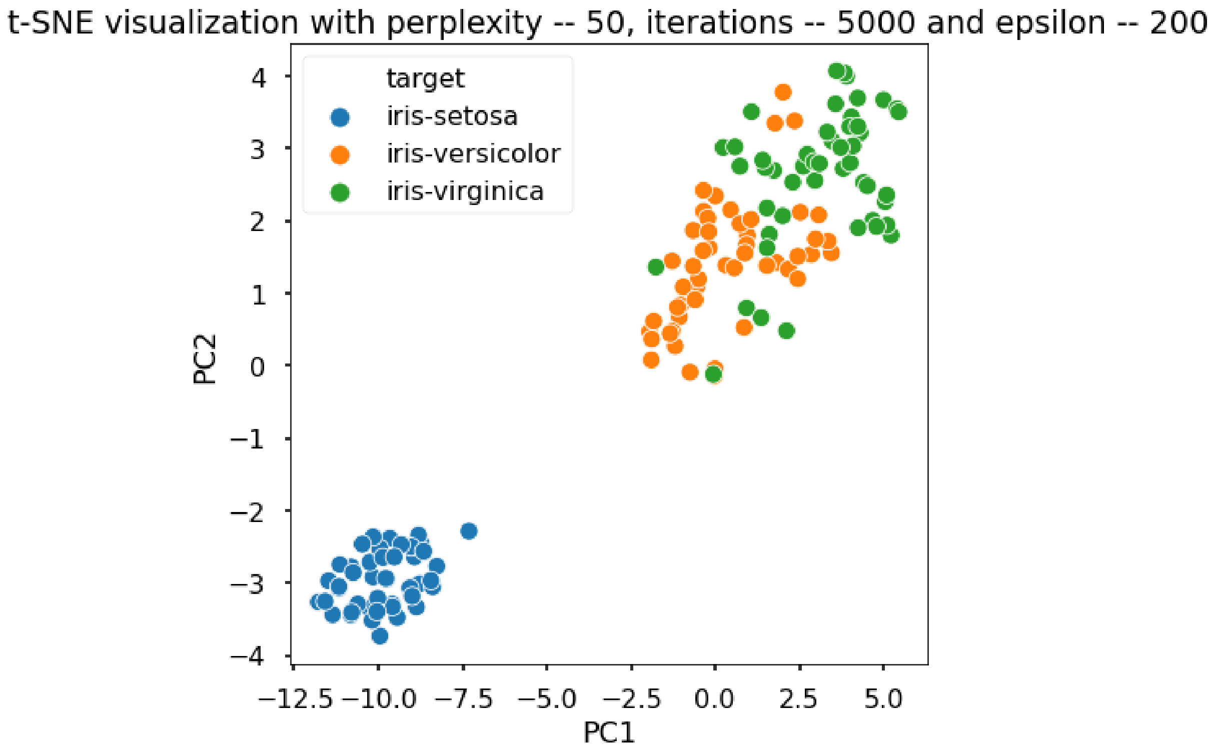
t-SNE visualization with perplexity -- 50, iterations -- 2500 and epsilon -- 200

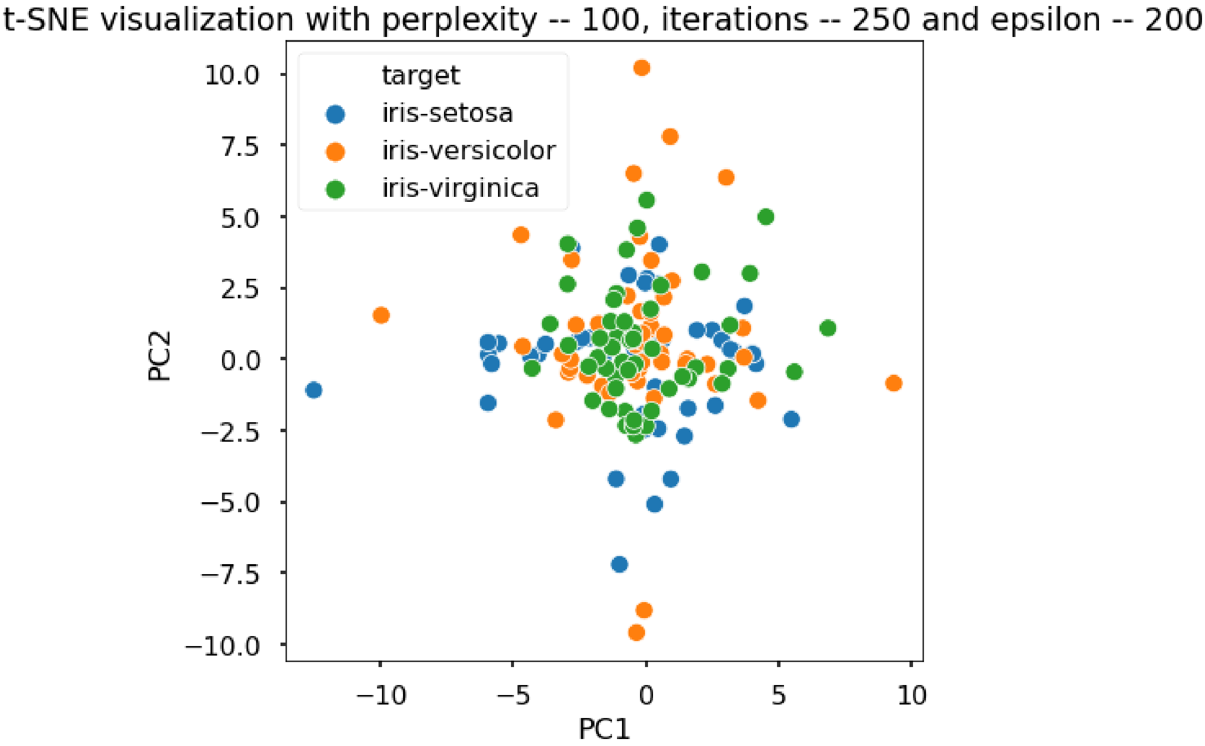


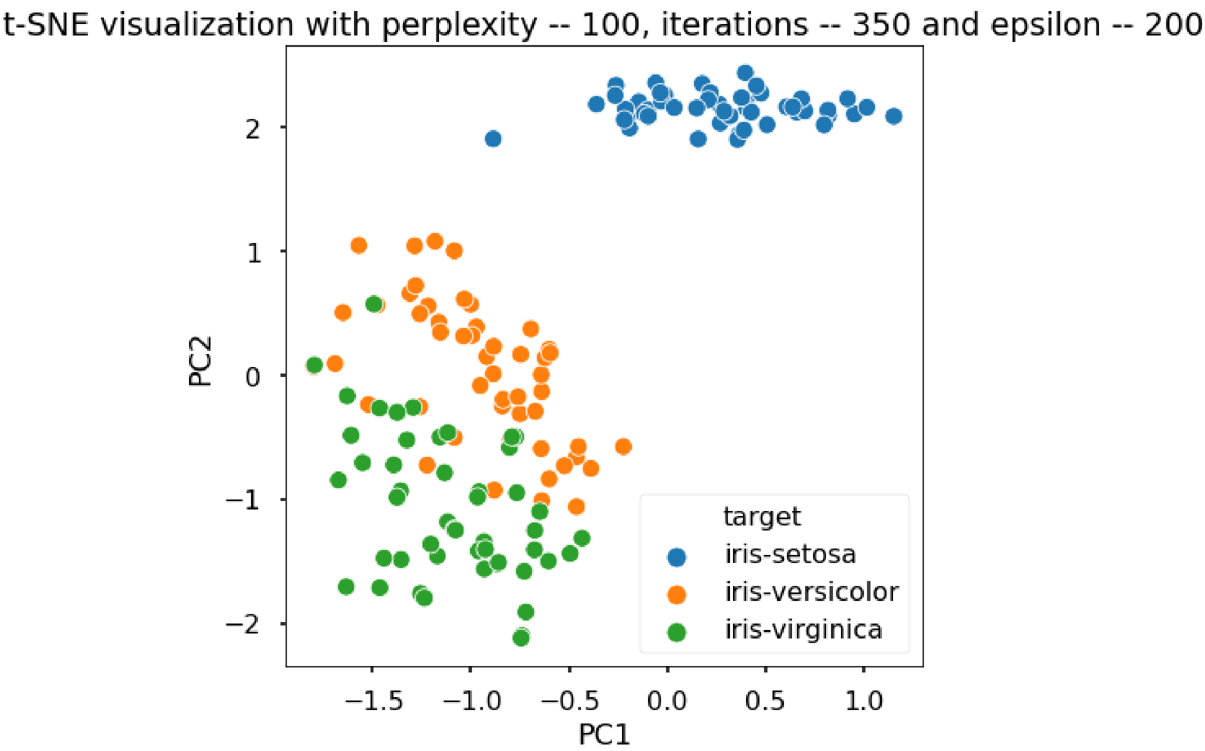
t-SNE visualization with perplexity -- 50, iterations -- 3500 and epsilon -- 200



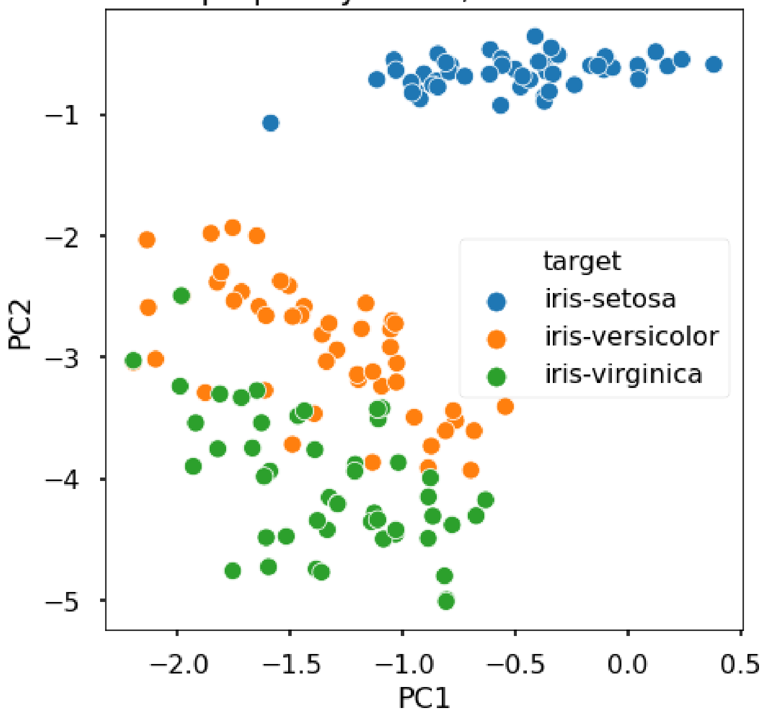

t-SNE visualization with perplexity -- 50, iterations -- 4000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
4
3
2
1
0
−1
−2
−3
−4
−12.5 −10.0 −7.5 −5.0 −2.5 0.0 2.5 5.0


t-SNE visualization with perplexity -- 50, iterations -- 5000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
4
3
2
1
0
−1
−2
−3
−4
−12.5 −10.0 −7.5 −5.0 −2.5 0.0 2.5 5.0

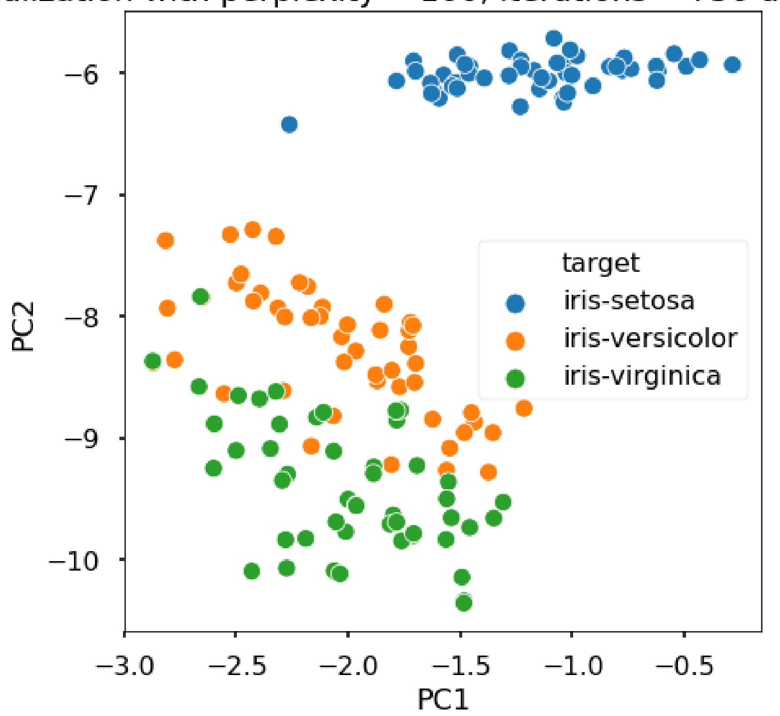

t-SNE visualization with perplexity -- 100, iterations -- 250 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC1
PC2

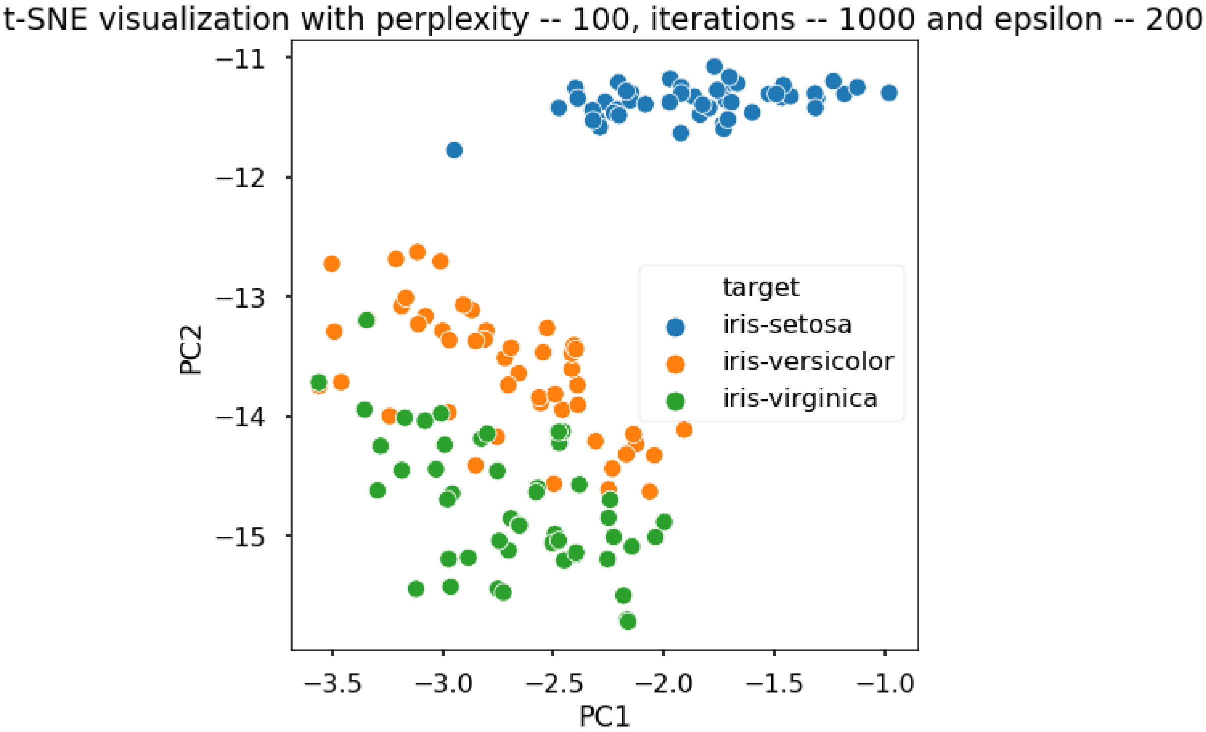

t-SNE visualization with perplexity -- 100, iterations -- 350 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC1
PC2

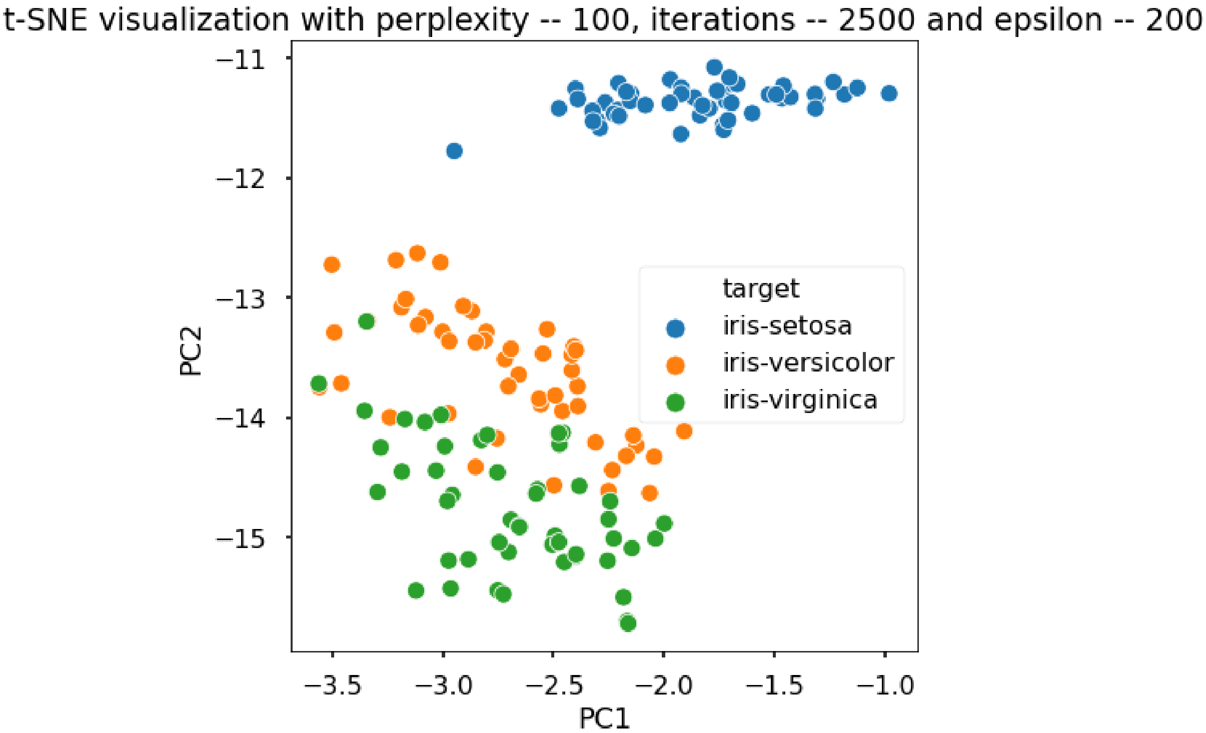
t-SNE visualization with perplexity -- 100, iterations -- 500 and epsilon -- 200

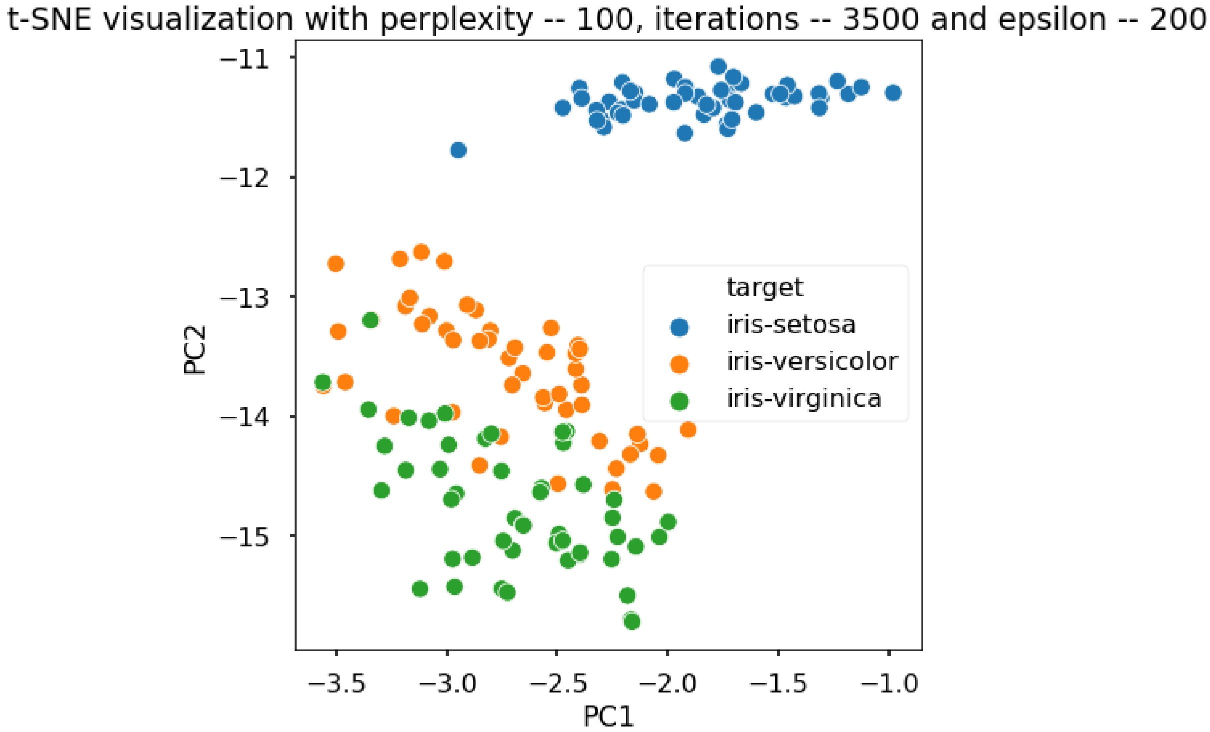


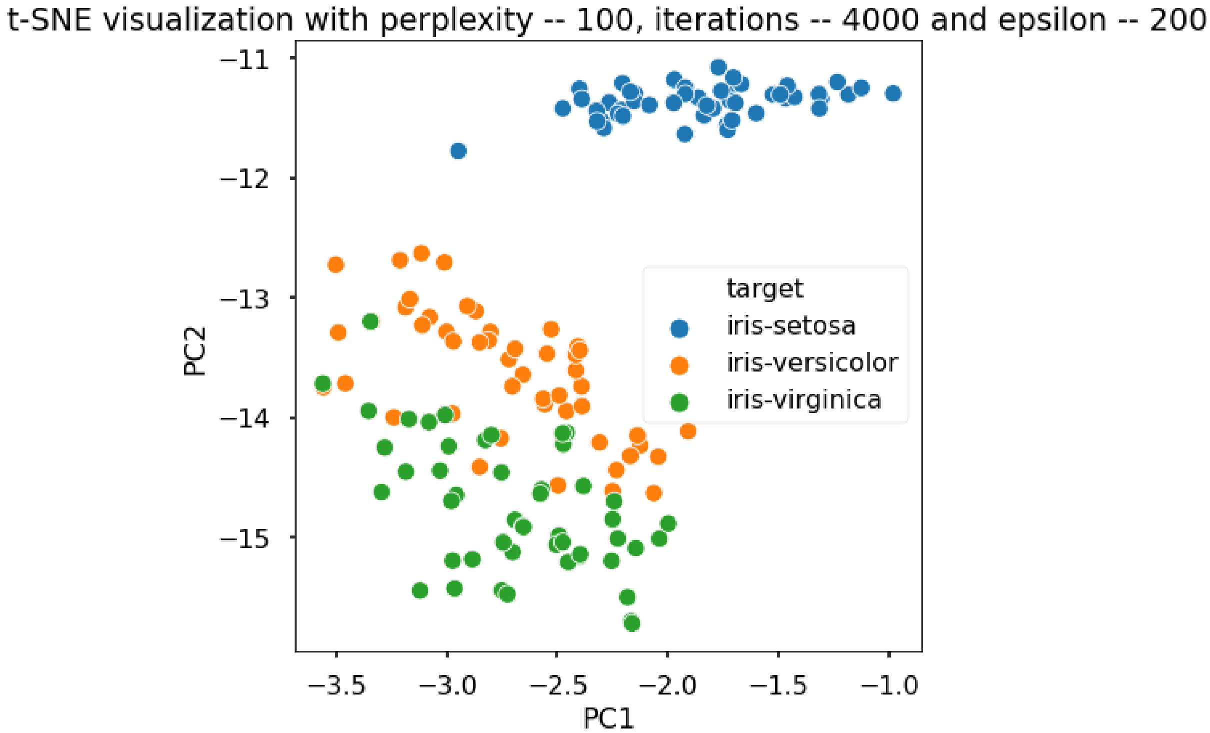
t-SNE visualization with perplexity -- 100, iterations -- 750 and epsilon -- 200



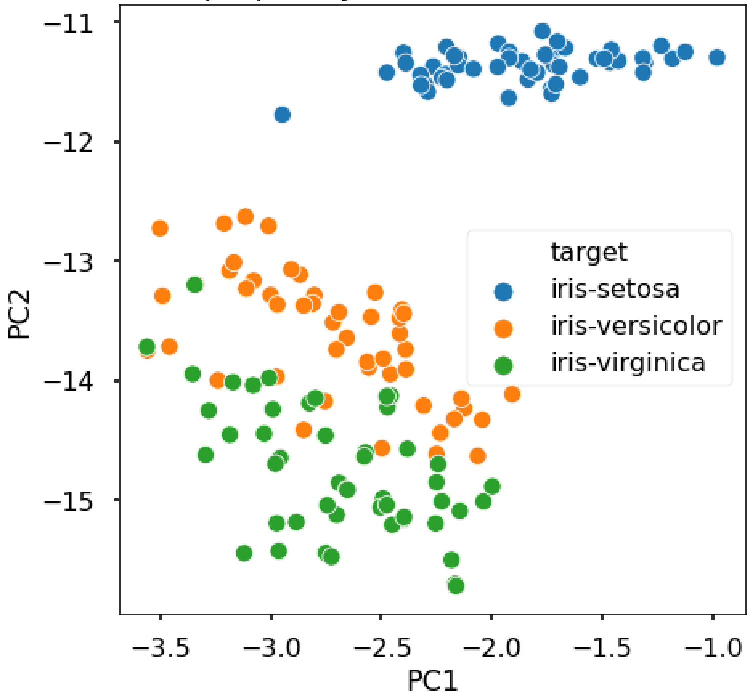

t-SNE visualization with perplexity -- 100, iterations -- 1000 and epsilon -- 200
−11
−12
−13
−14
−15
PC2
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
PC1
target
iris-setosa
iris-versicolor
iris-virginica


t-SNE visualization with perplexity -- 100, iterations -- 2500 and epsilon -- 200
−11
−12
−13
−14
−15
PC2
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
PC1
target
iris-setosa
iris-versicolor
iris-virginica

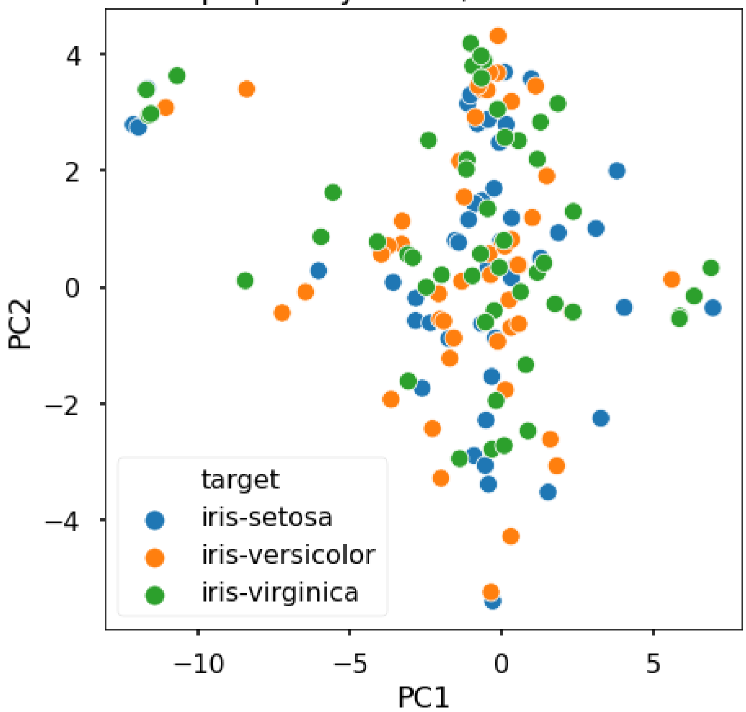

t-SNE visualization with perplexity -- 100, iterations -- 3500 and epsilon -- 200
−11
−12
−13
−14
−15
PC2
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
PC1
target
iris-setosa
iris-versicolor
iris-virginica

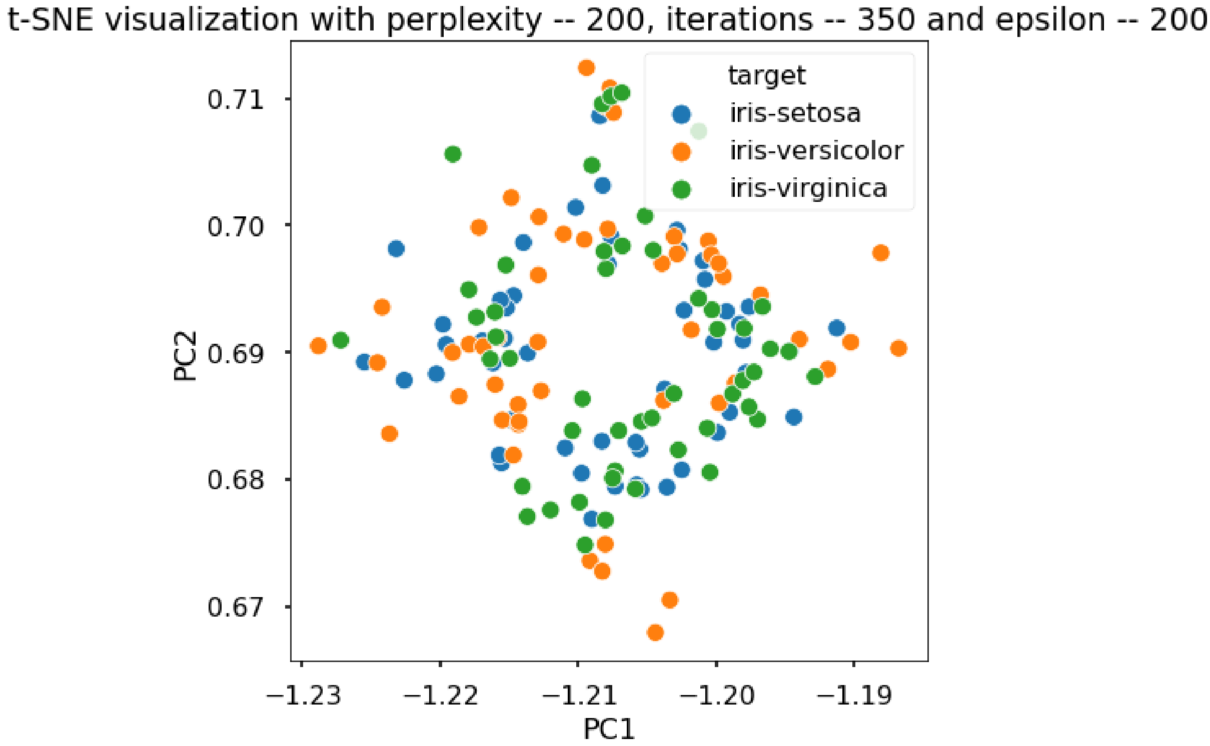

t-SNE visualization with perplexity -- 100, iterations -- 4000 and epsilon -- 200
−11
−12
−13
−14
−15
PC2
−3.5
−3.0
−2.5
−2.0
−1.5
−1.0
PC1
target
iris-setosa
iris-versicolor
iris-virginica

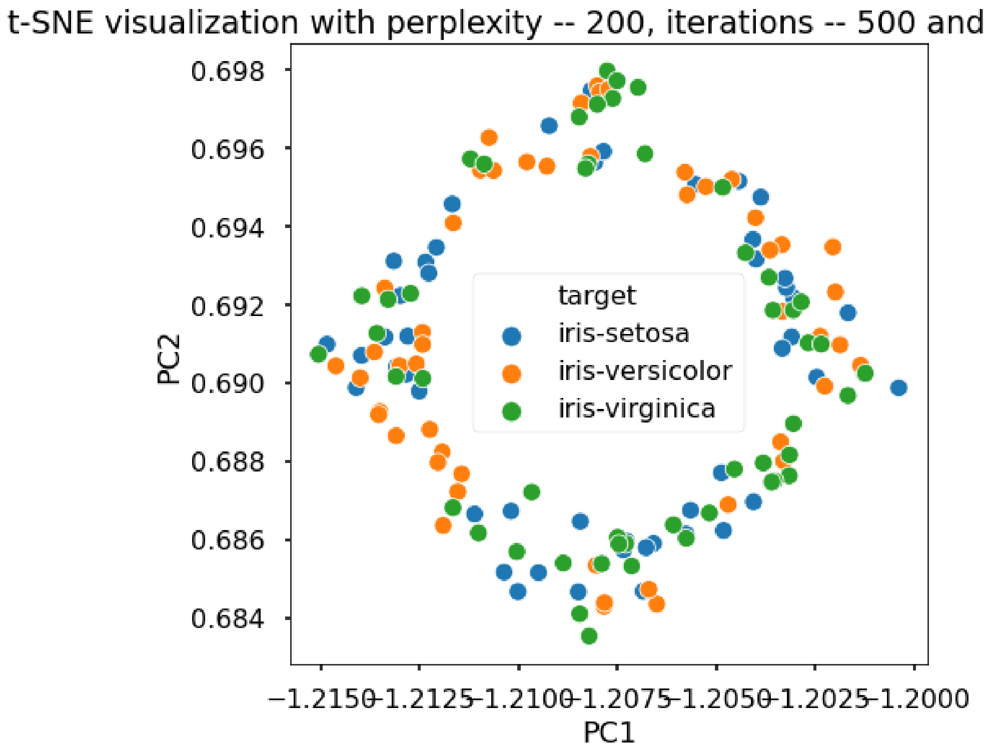
t-SNE visualization with perplexity -- 100, iterations -- 5000 and epsilon -- 200

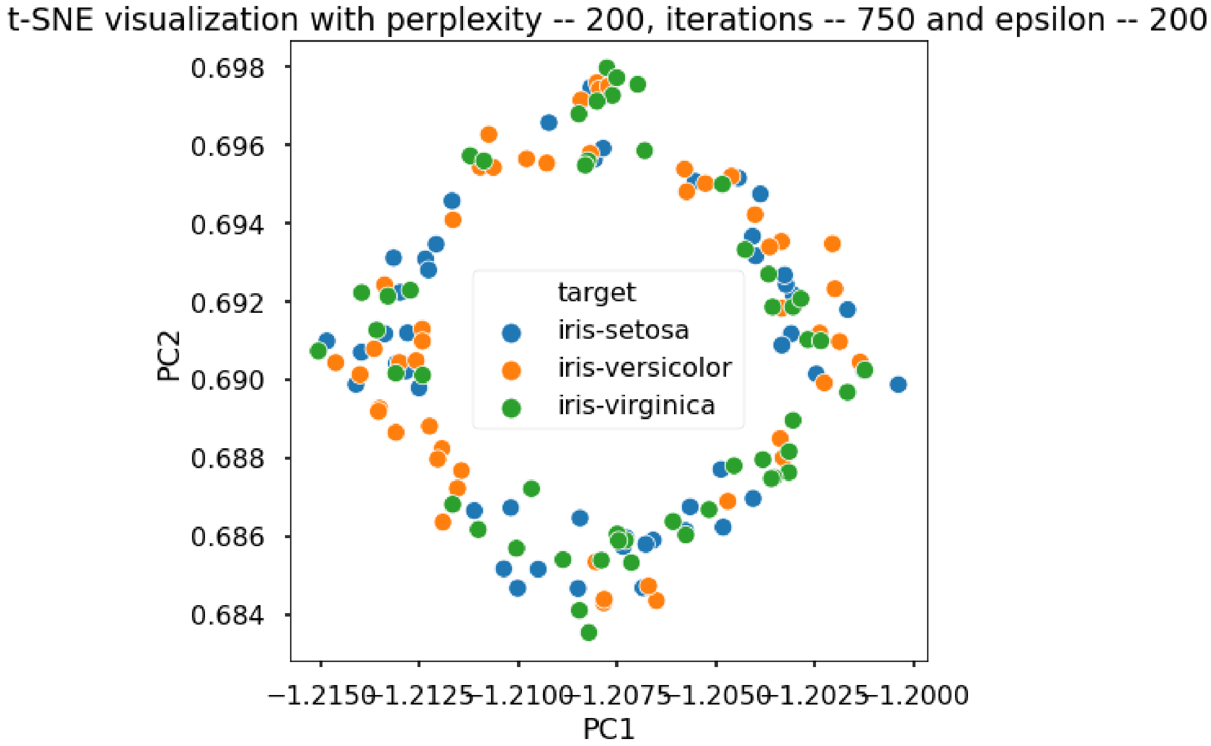


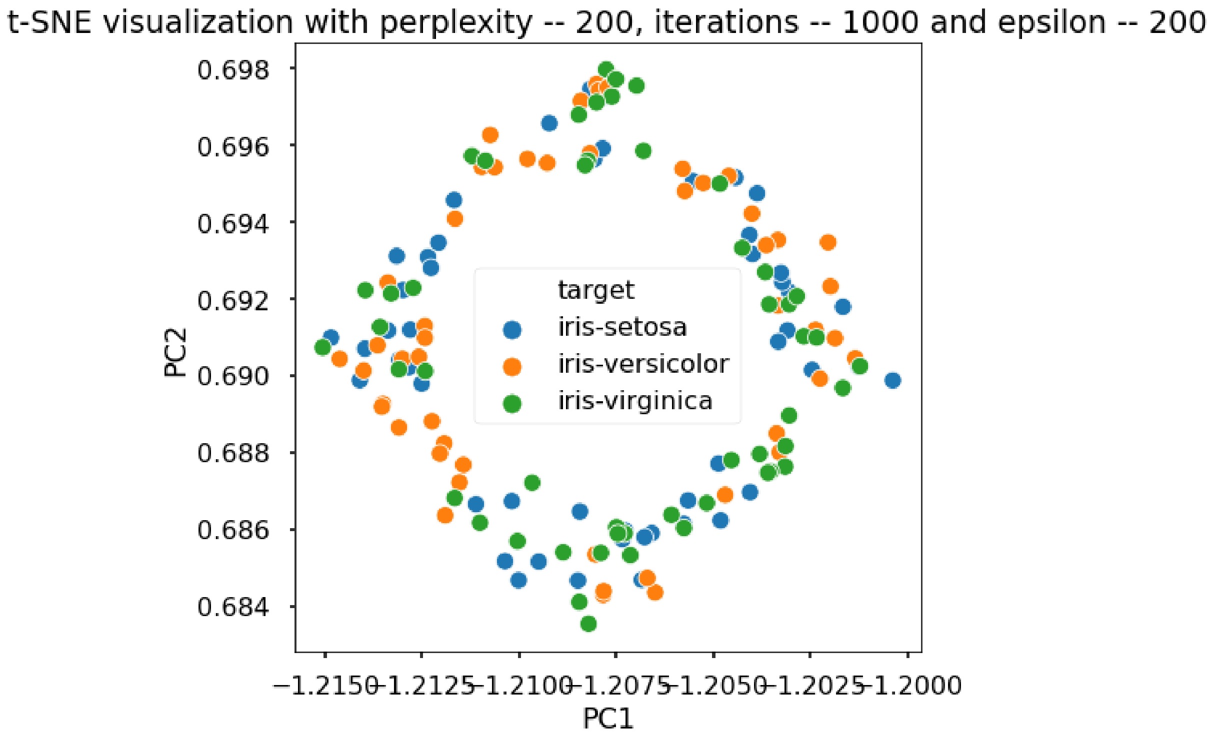
t-SNE visualization with perplexity -- 200, iterations -- 250 and epsilon -- 200

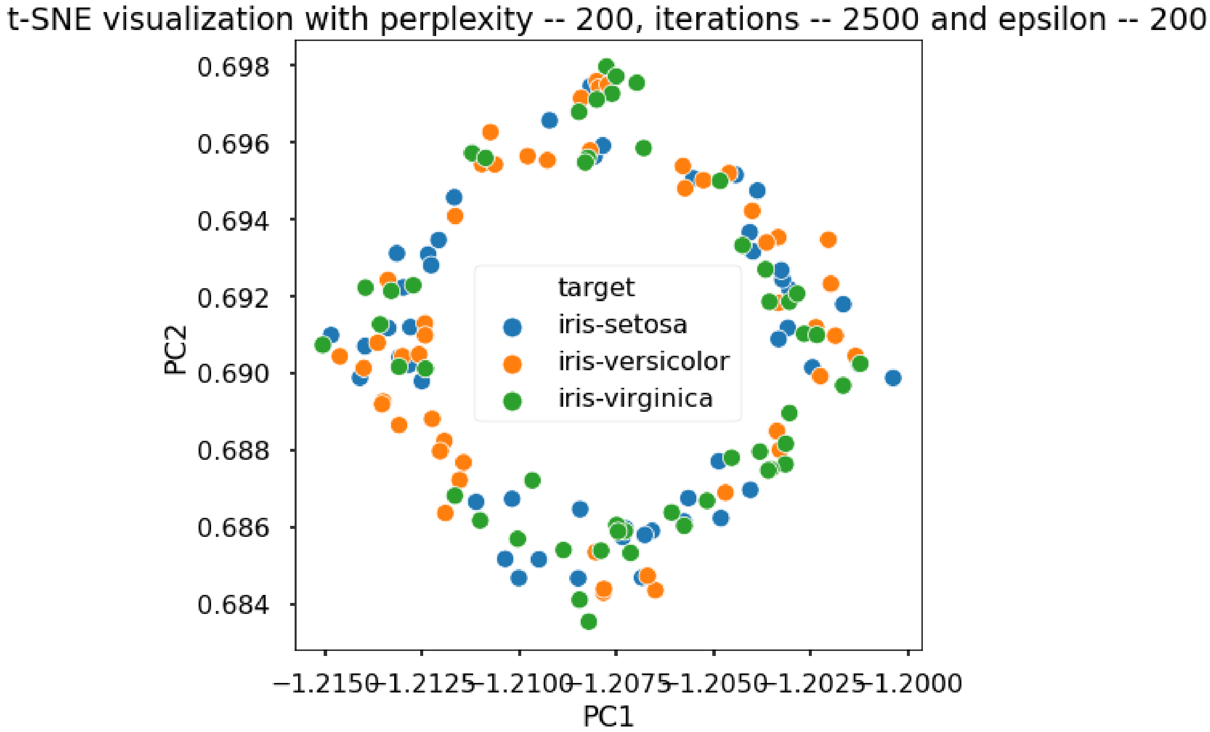


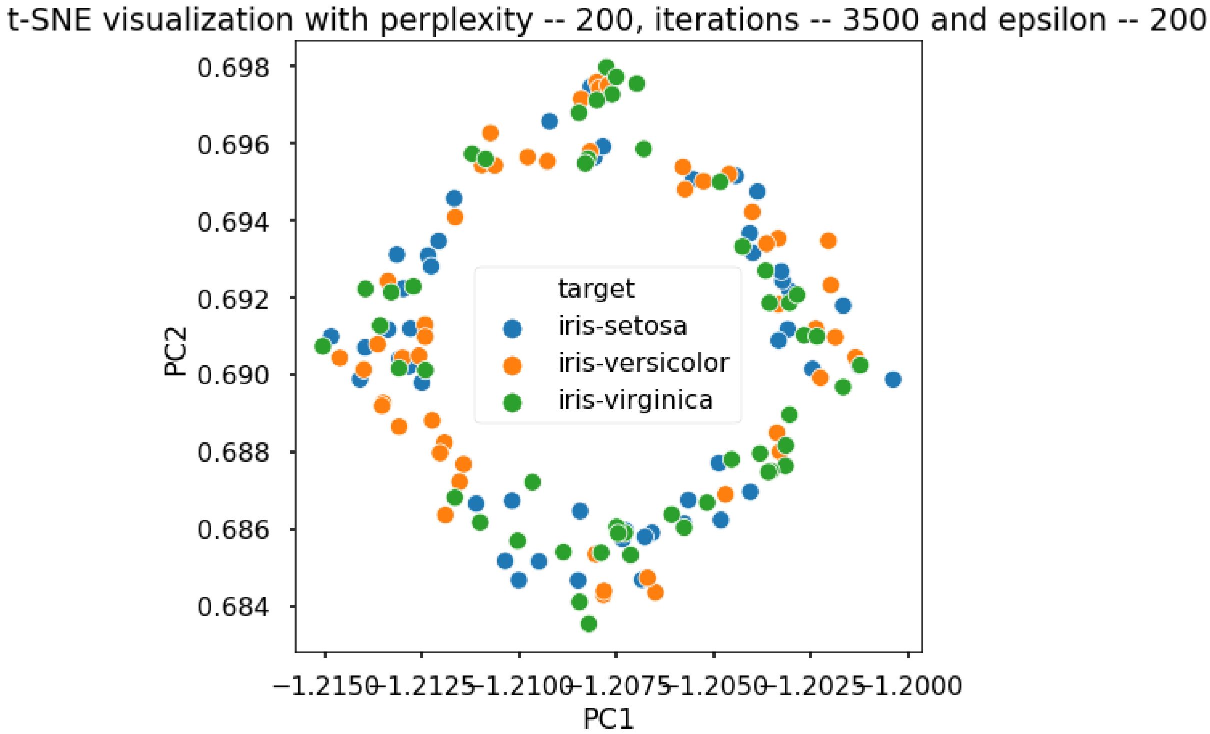

t-SNE visualization with perplexity -- 200, iterations -- 350 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
0.71
0.70
0.69
0.68
0.67
PC2
−1.23
−1.22
−1.21
−1.20
−1.19
PC1


t-SNE visualization with perplexity -- 200, iterations -- 500 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
0.698
0.696
0.694
0.692
0.690
0.688
0.686
0.684
PC2
−1.2150
−1.2125
−1.2100
−1.2075
−1.2050
−1.2025
−1.2000
PC1

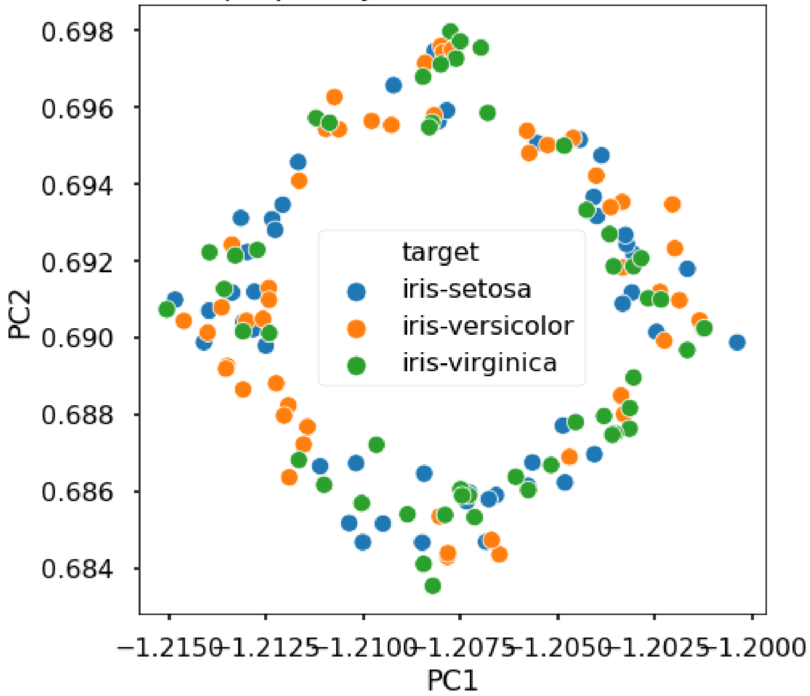

t-SNE visualization with perplexity -- 200, iterations -- 750 and epsilon -- 200
0.698
0.696
0.694
0.692
0.690
0.688
0.686
0.684
PC2
target
iris-setosa
iris-versicolor
iris-virginica
−1.2150
−1.2125
−1.2100
−1.2075
−1.2050
−1.2025
−1.2000
PC1


t-SNE visualization with perplexity -- 200, iterations -- 1000 and epsilon -- 200
0.698
0.696
0.694
0.692
0.690
0.688
0.686
0.684
PC2
target
iris-setosa
iris-versicolor
iris-virginica
−1.2150
−1.2125
−1.2100
−1.2075
−1.2050
−1.2025
−1.2000
PC1

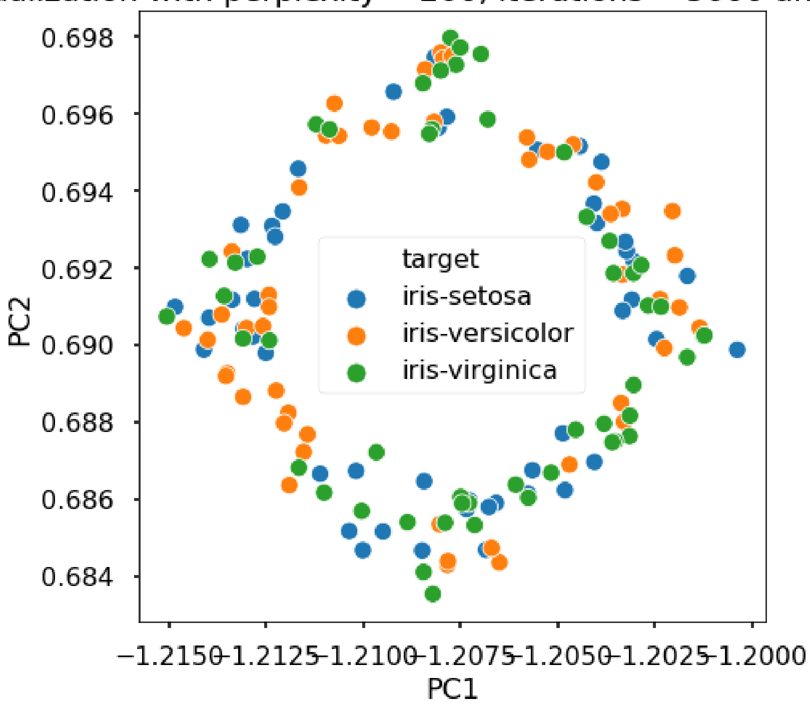

t-SNE visualization with perplexity -- 200, iterations -- 2500 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
0.698
0.696
0.694
0.692
0.690
0.688
0.686
0.684
−1.2150
−1.2125
−1.2100
−1.2075
−1.2050
−1.2025
−1.2000
PC1


t-SNE visualization with perplexity -- 200, iterations -- 3500 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
0.698
0.696
0.694
0.692
0.690
0.688
0.686
0.684
−1.2150
−1.2125
−1.2100
−1.2075
−1.2050
−1.2025
−1.2000
PC1

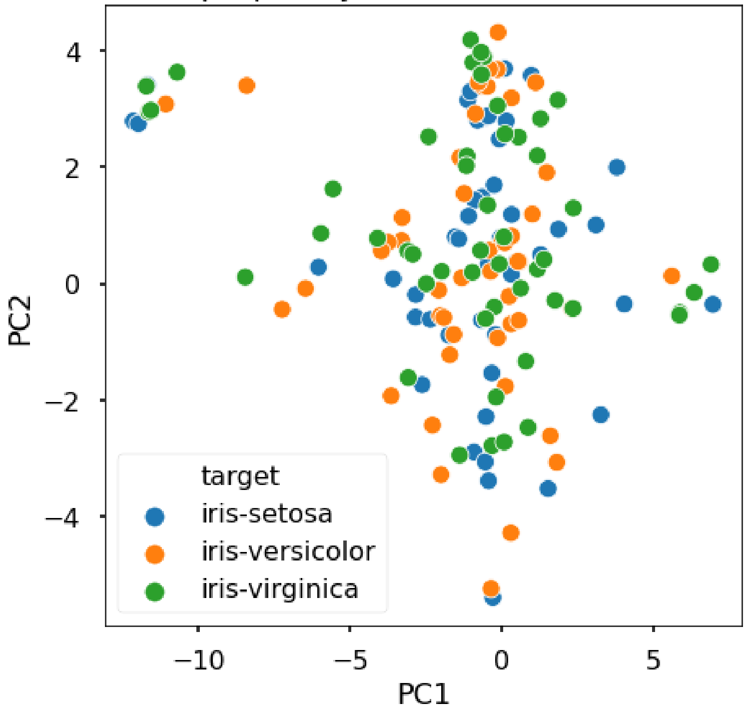
t-SNE visualization with perplexity -- 200, iterations -- 4000 and epsilon -- 200



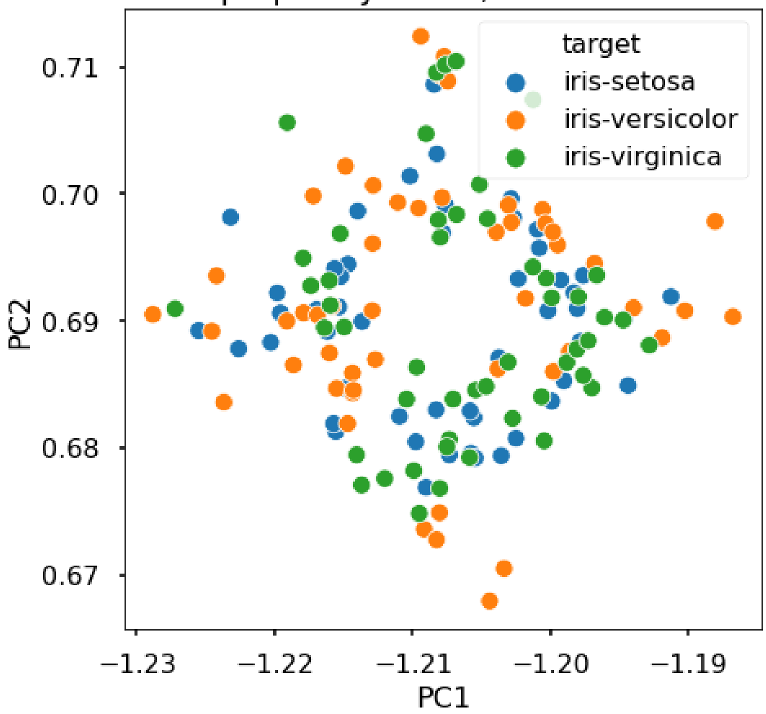
t-SNE visualization with perplexity -- 200, iterations -- 5000 and epsilon -- 200

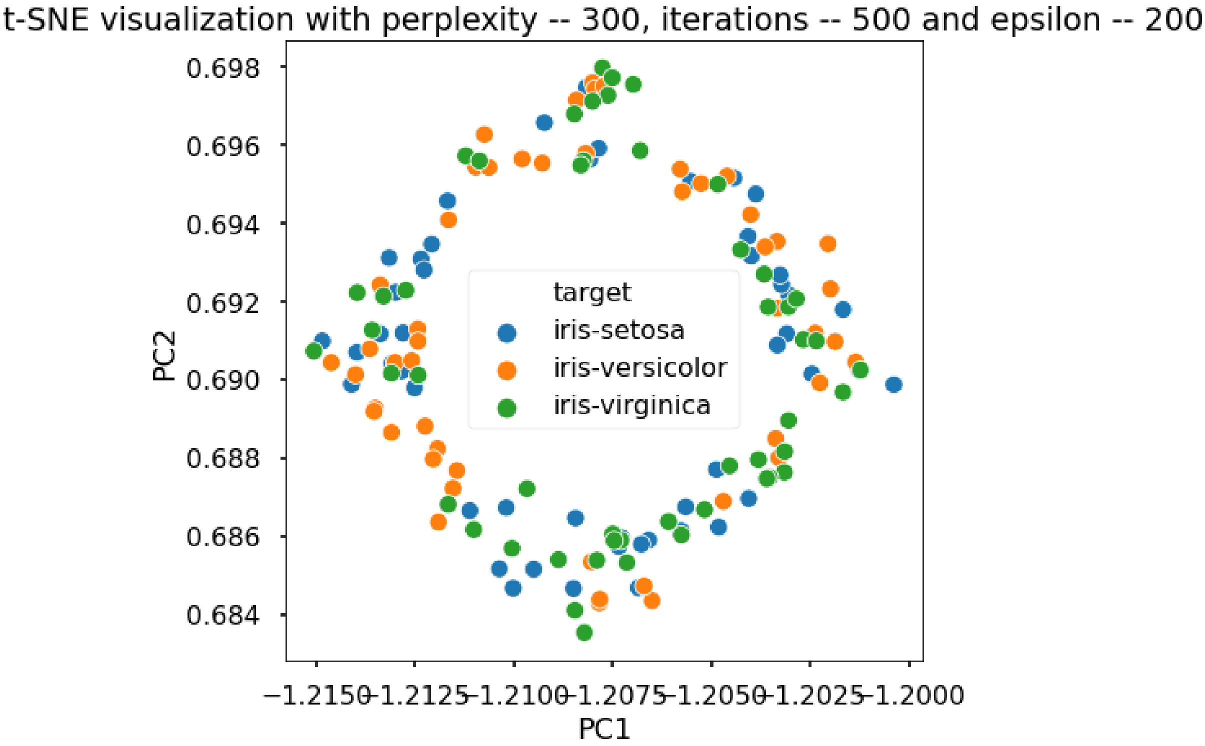


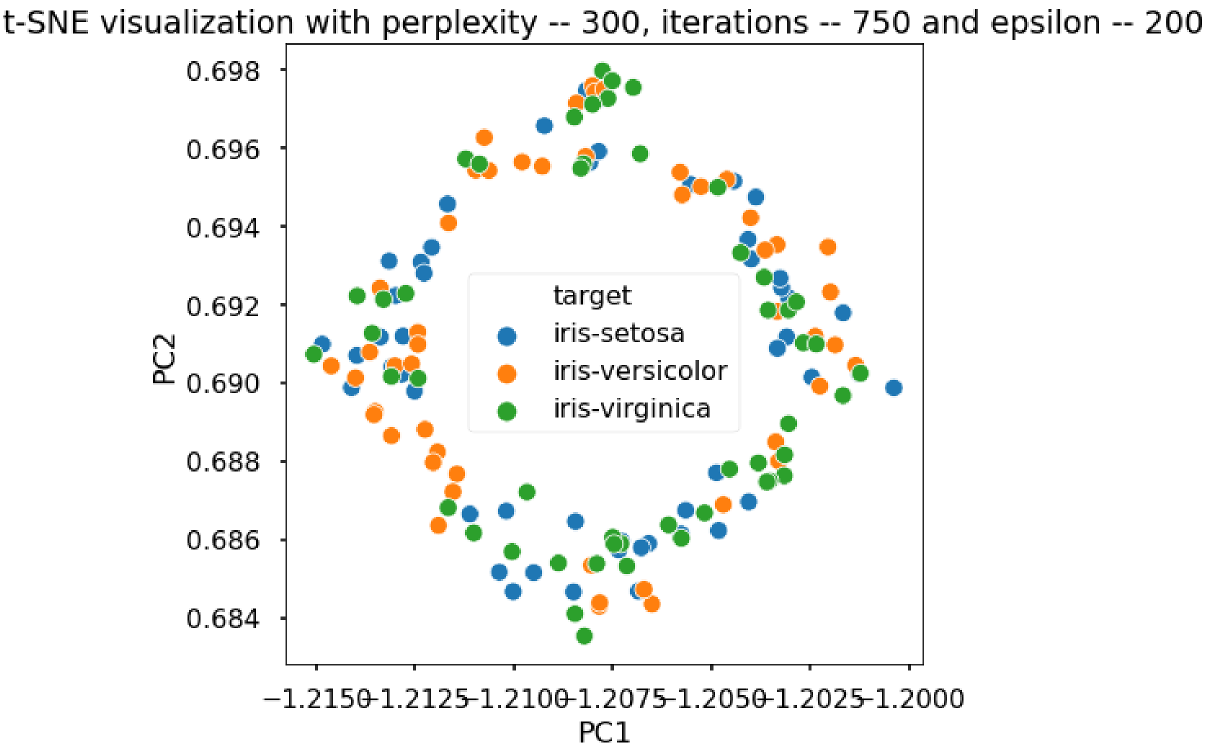
t-SNE visualization with perplexity -- 300, iterations -- 250 and epsilon -- 200

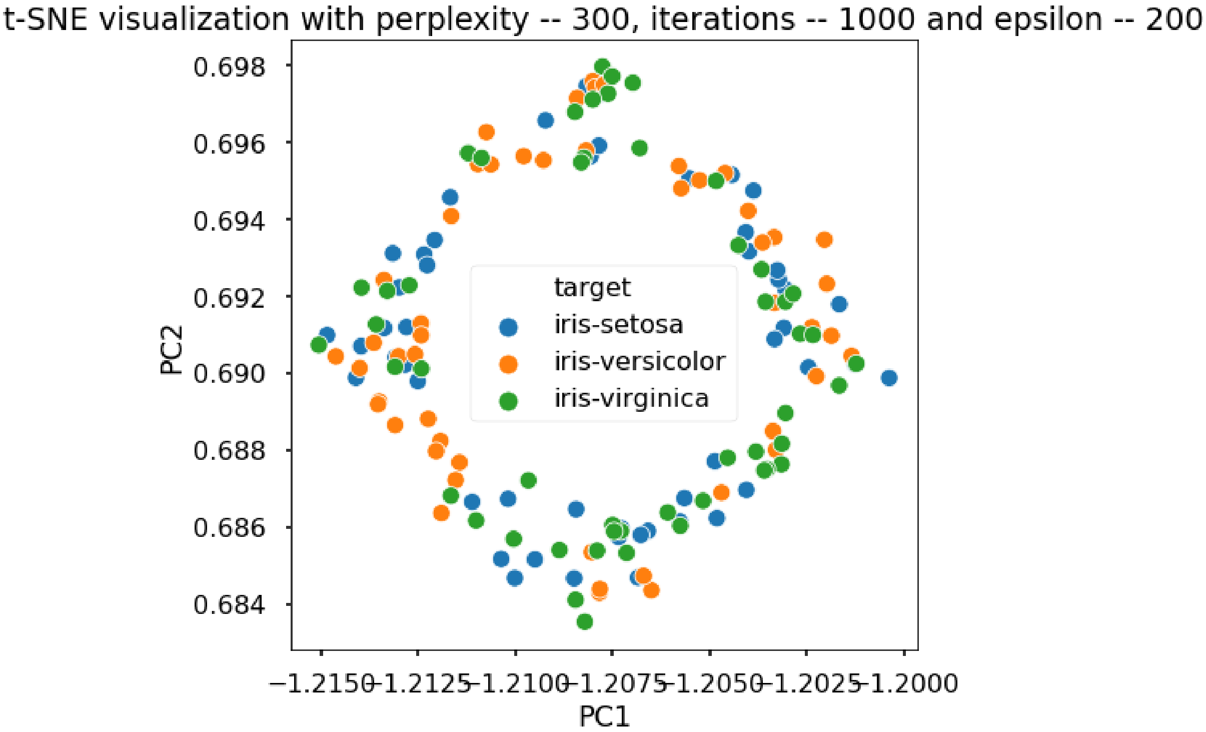


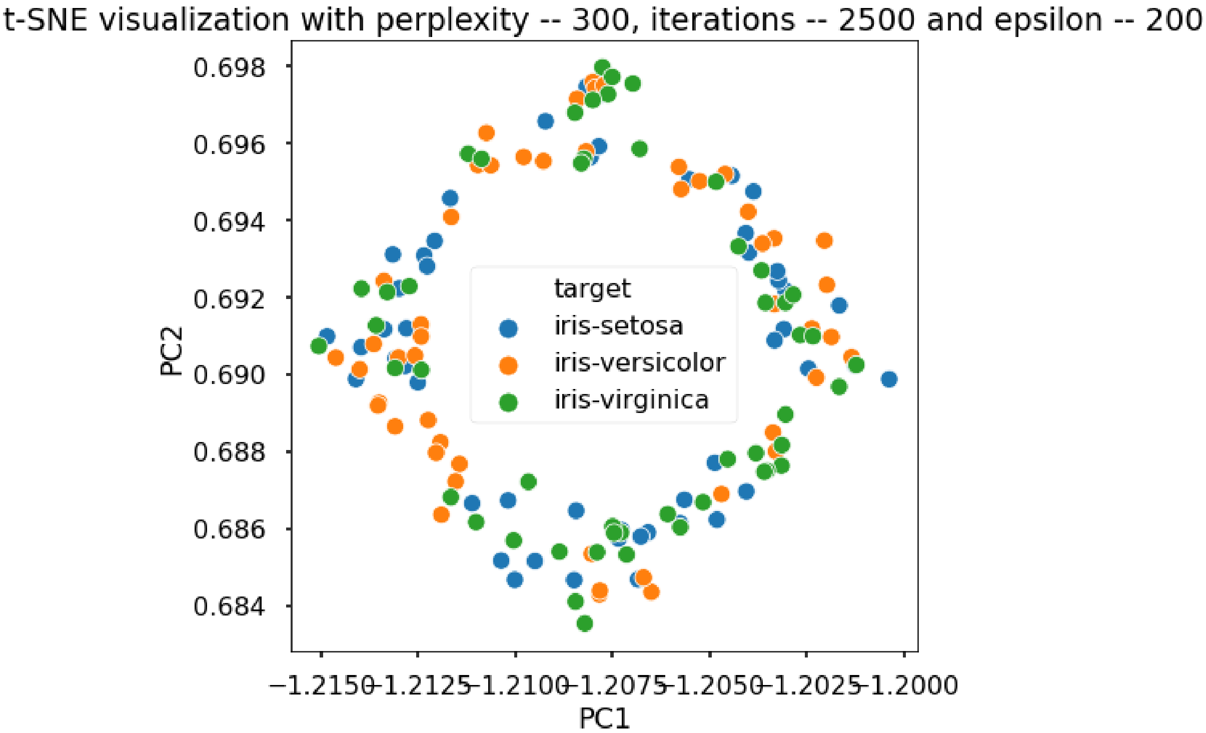
t-SNE visualization with perplexity -- 300, iterations -- 350 and epsilon -- 200

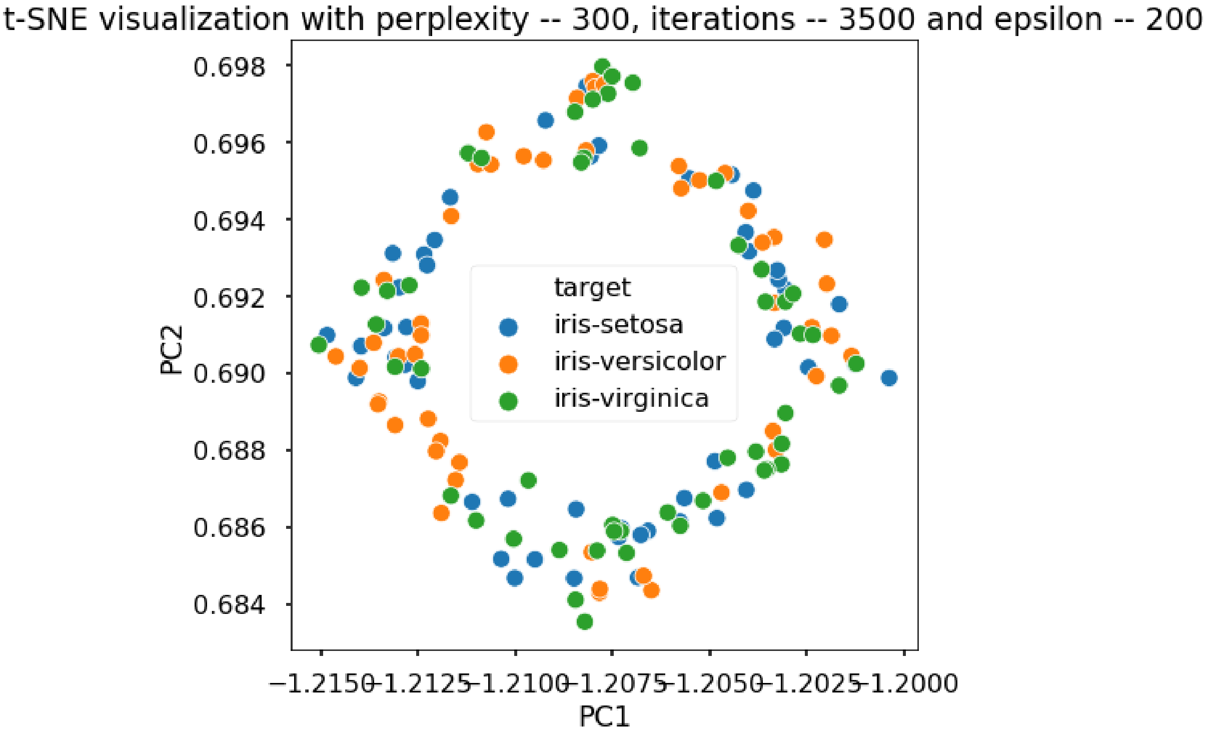


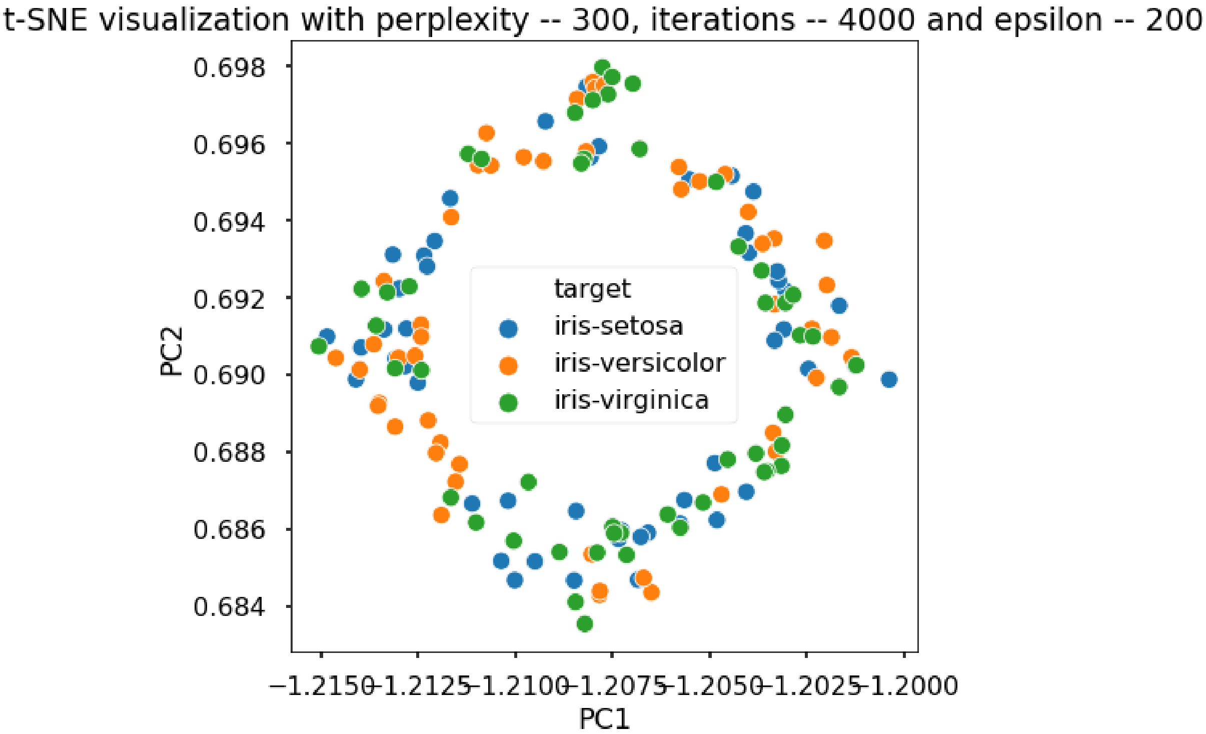

t-SNE visualization with perplexity -- 300, iterations -- 500 and epsilon -- 200
0.698
0.696
0.694
0.692
0.690
0.688
0.686
0.684
PC2
target
iris-setosa
iris-versicolor
iris-virginica
−1.2150
−1.2125
−1.2100
−1.2075
−1.2050
−1.2025
−1.2000
PC1

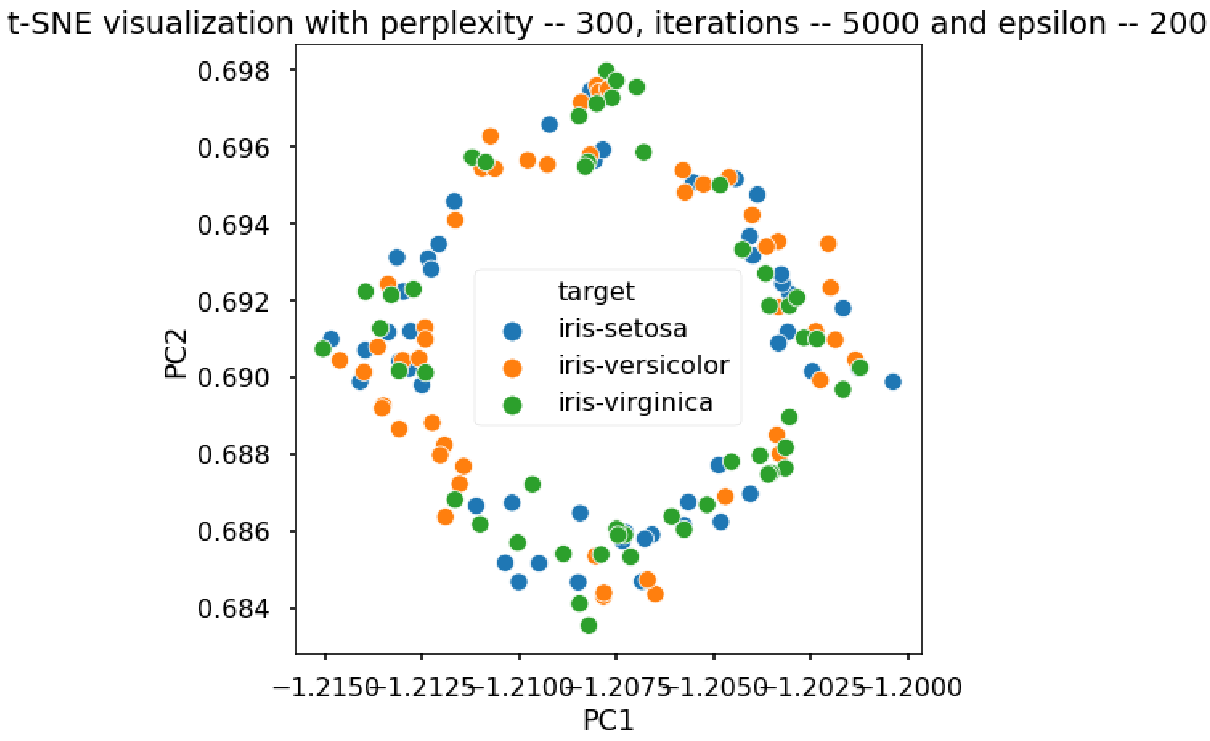

t-SNE visualization with perplexity -- 300, iterations -- 750 and epsilon -- 200
0.698
0.696
0.694
0.692
0.690
0.688
0.686
0.684
PC2
target
iris-setosa
iris-versicolor
iris-virginica
−1.2150
−1.2125
−1.2100
−1.2075
−1.2050
−1.2025
−1.2000
PC1


t-SNE visualization with perplexity -- 300, iterations -- 1000 and epsilon -- 200
PC2
0.698
0.696
0.694
0.692
0.690
0.688
0.686
0.684
target
iris-setosa
iris-versicolor
iris-virginica
−1.2150
−1.2125
−1.2100
−1.2075
−1.2050
−1.2025
−1.2000
PC1


t-SNE visualization with perplexity -- 300, iterations -- 2500 and epsilon -- 200
PC2
0.698
0.696
0.694
0.692
0.690
0.688
0.686
0.684
target
iris-setosa
iris-versicolor
iris-virginica
−1.2150
−1.2125
−1.2100
−1.2075
−1.2050
−1.2025
−1.2000
PC1


t-SNE visualization with perplexity -- 300, iterations -- 3500 and epsilon -- 200
PC2
PC1
0.698
0.696
0.694
0.692
0.690
0.688
0.686
0.684
−1.2150
−1.2125
−1.2100
−1.2075
−1.2050
−1.2025
−1.2000
target
iris-setosa
iris-versicolor
iris-virginica


t-SNE visualization with perplexity -- 300, iterations -- 4000 and epsilon -- 200
PC2
PC1
0.698
0.696
0.694
0.692
0.690
0.688
0.686
0.684
−1.2150
−1.2125
−1.2100
−1.2075
−1.2050
−1.2025
−1.2000
target
iris-setosa
iris-versicolor
iris-virginica



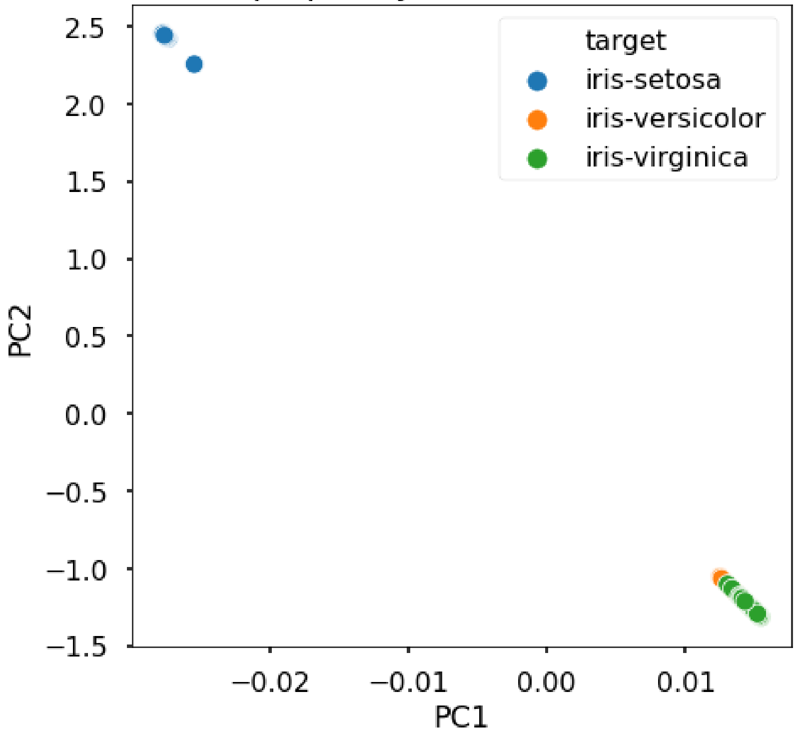
IRIS:CASE-II-Multiple_runs

Running t-SNE on a range of learning rates(epsilon) and iterations with fixed perplexity and embedding initialization as 'random'

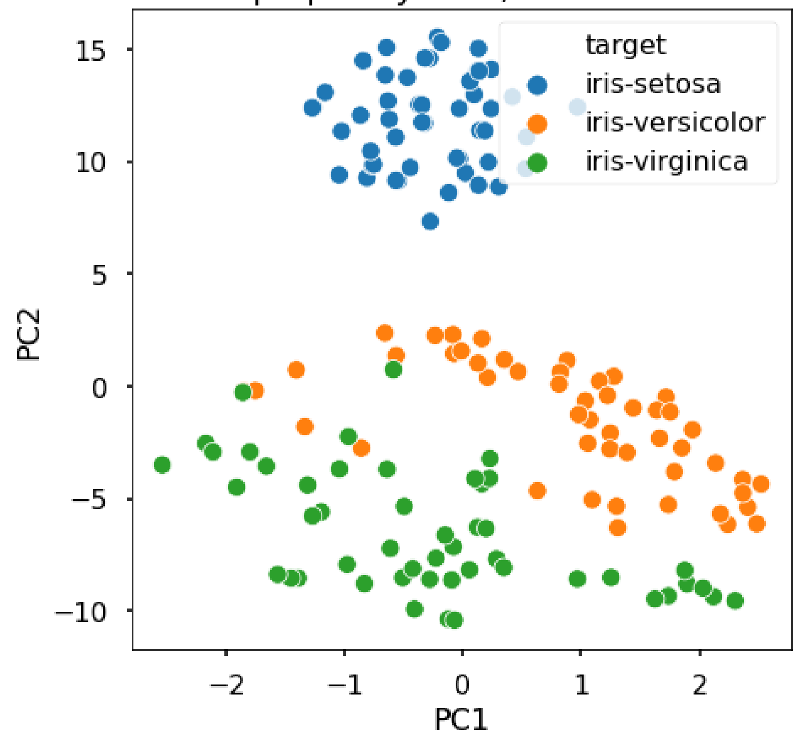
In [79]:

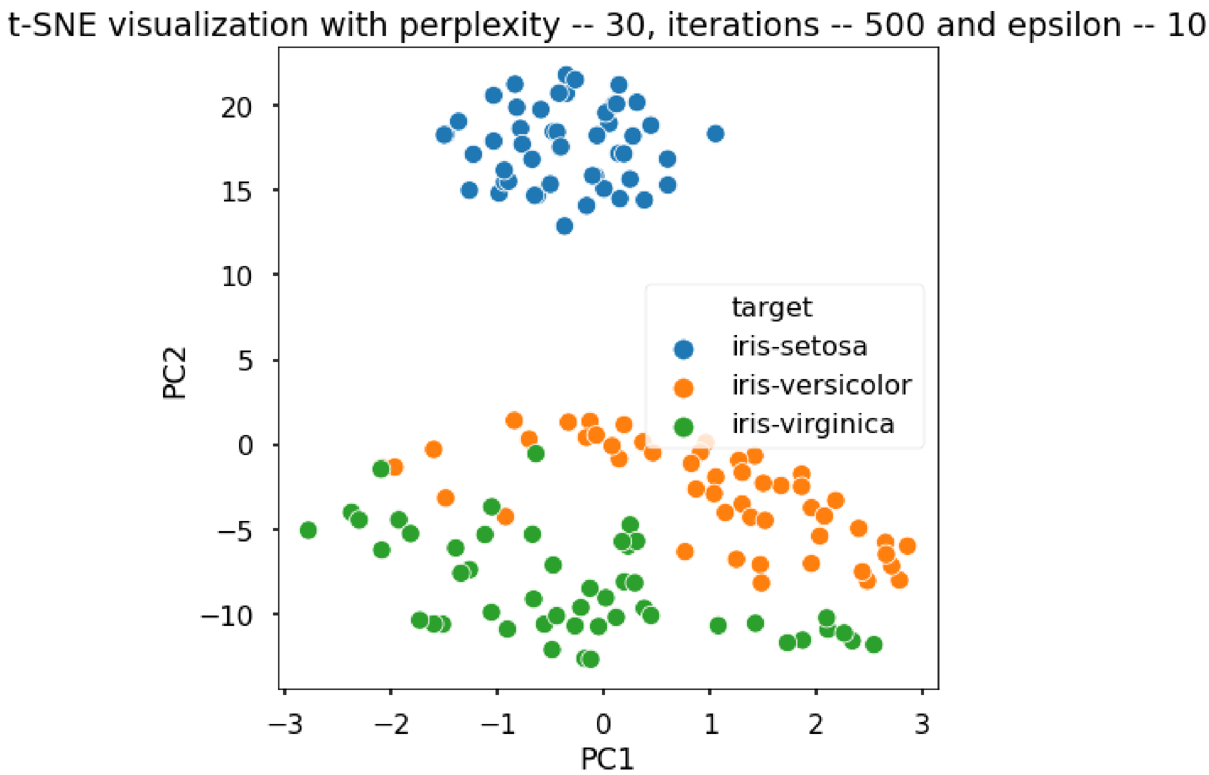
```
l_rate = [10,30,50,100,200,400,600]
iterations = [250,350,500,750,1000,2500,3500,4000,5000]

for lidx in range(len(l_rate)):
    for idx in range(len(iterations)):
        tsne_iris_1 = TSNE(n_components=2,perplexity=30,learning_rate=l_rate[lidx],n
                               init='random',n_jobs=-1,random_state=23)
        iris1_tsne_pcmps = pd.DataFrame(tsne_iris_1.fit_transform(iris_stand_df.iloc
        iris1_tsne_pcmps = pd.concat([iris1_tsne_pcmps,iris_stand_df['target']],axis
        with plt.style.context('seaborn-poster'):
            plt.figure(figsize=(7,7))
            sns.scatterplot(data=iris1_tsne_pcmps,x='PC1',y='PC2',hue='target')
            plt.title("t-SNE visualization with perplexity -- {0}, iterations -- {1}


            plt.show()
```

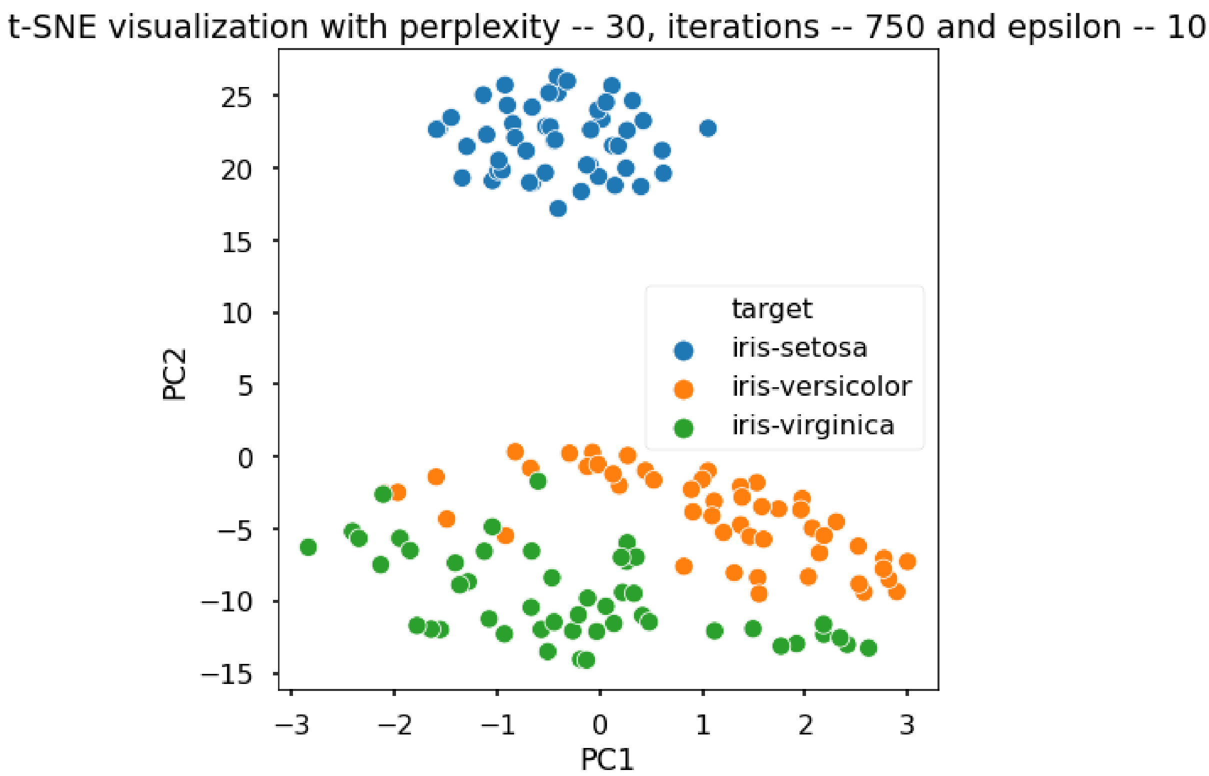
t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 10

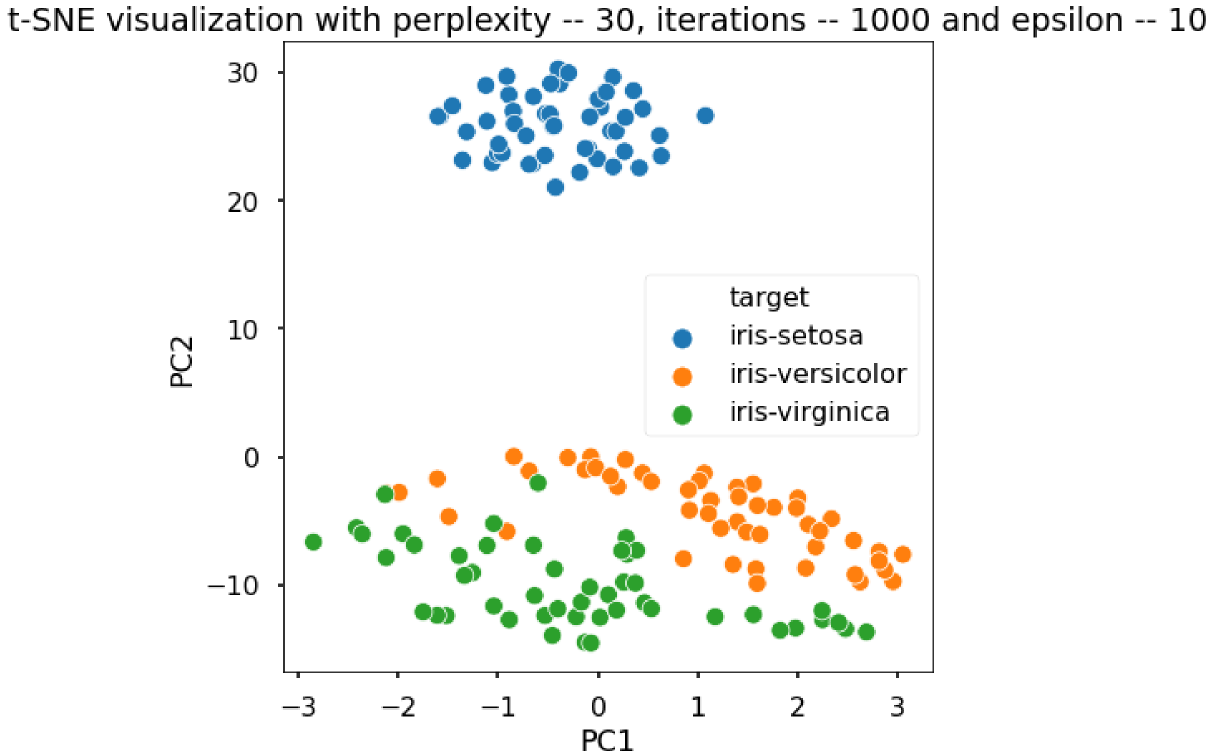


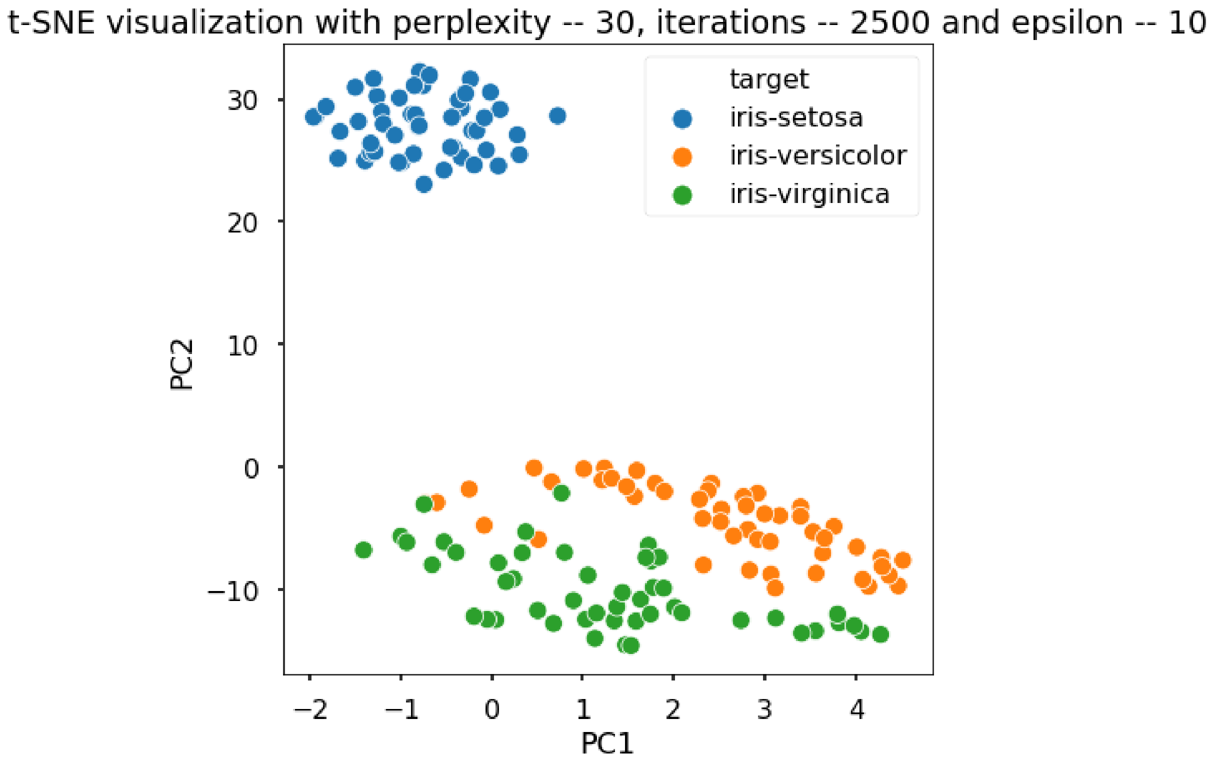
t-SNE visualization with perplexity -- 30, iterations -- 350 and epsilon -- 10

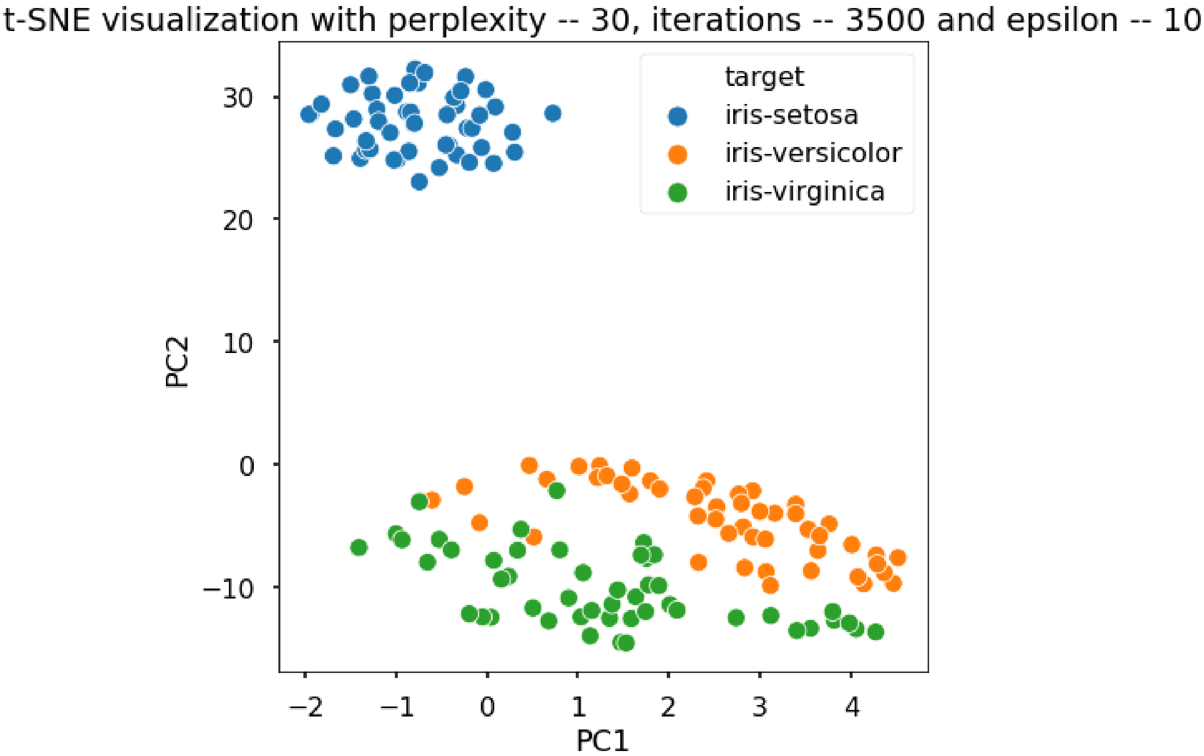


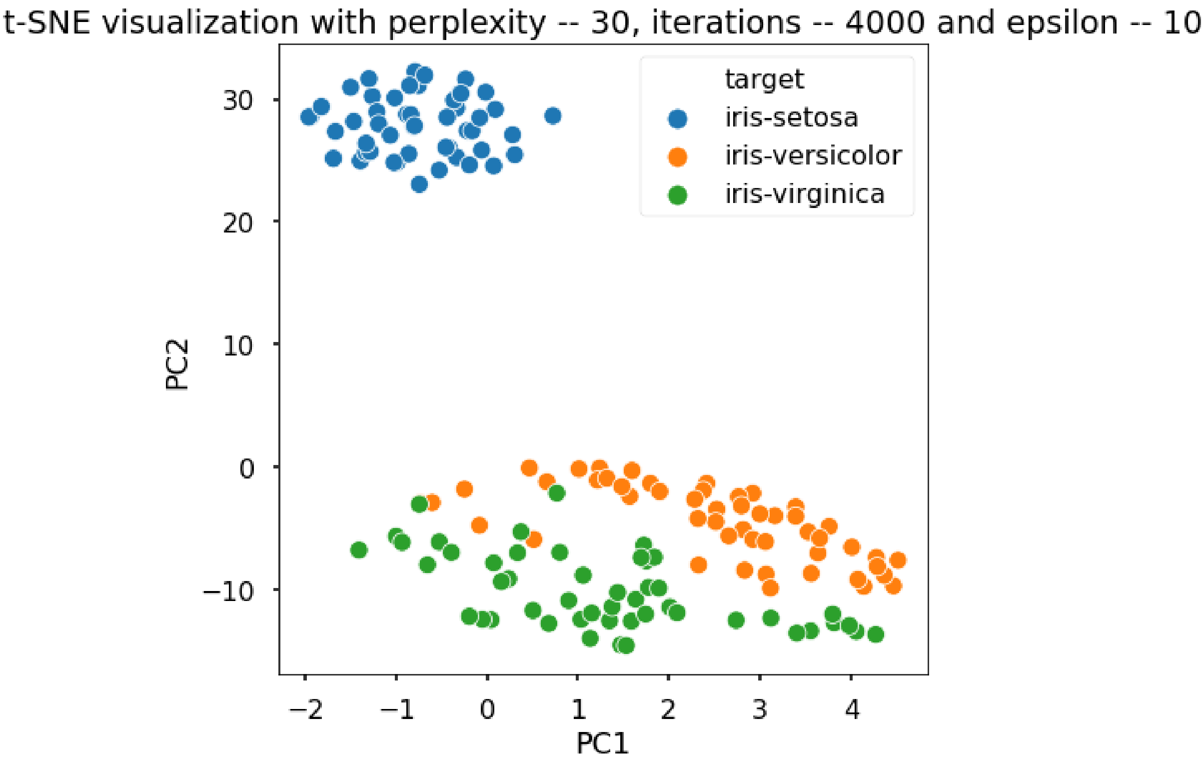

t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 10
PC2
PC1
target
iris-setosa
iris-versicolor
iris-virginica
20
15
10
5
0
−5
−10
−3
−2
−1
0
1
2
3

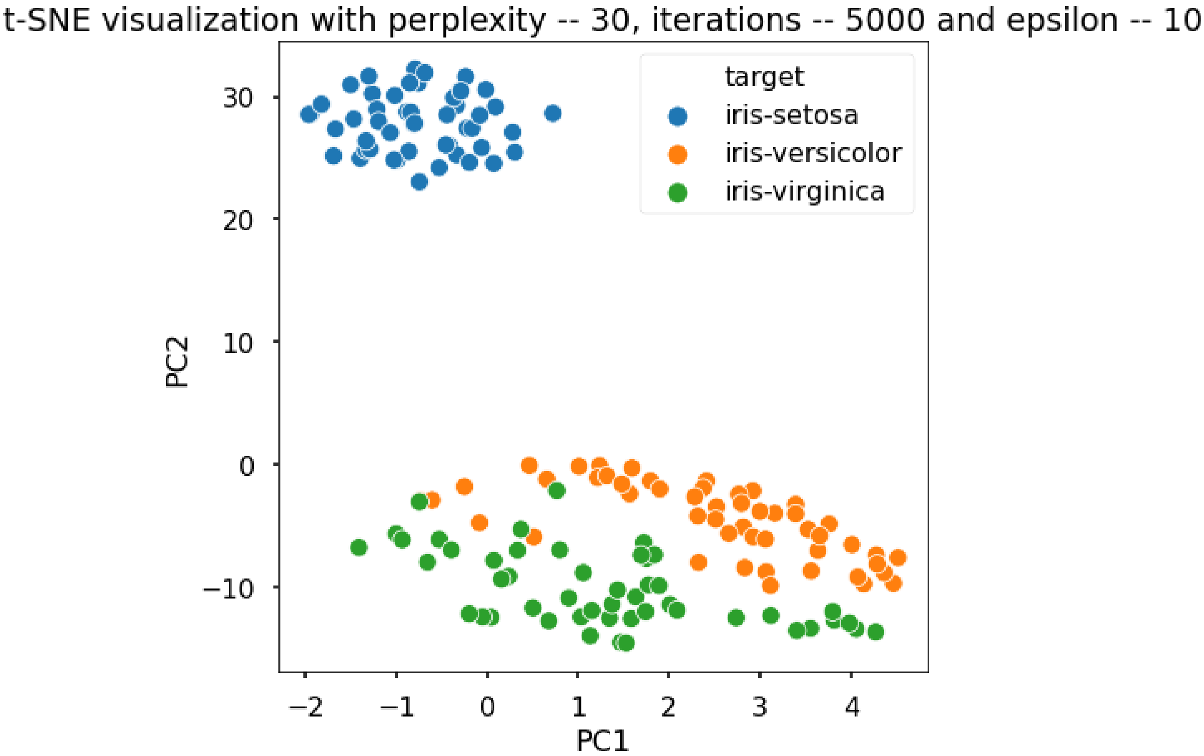

t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 10
PC2
PC1
target
iris-setosa
iris-versicolor
iris-virginica
25
20
15
10
5
0
−5
−10
−15
−3
−2
−1
0
1
2
3

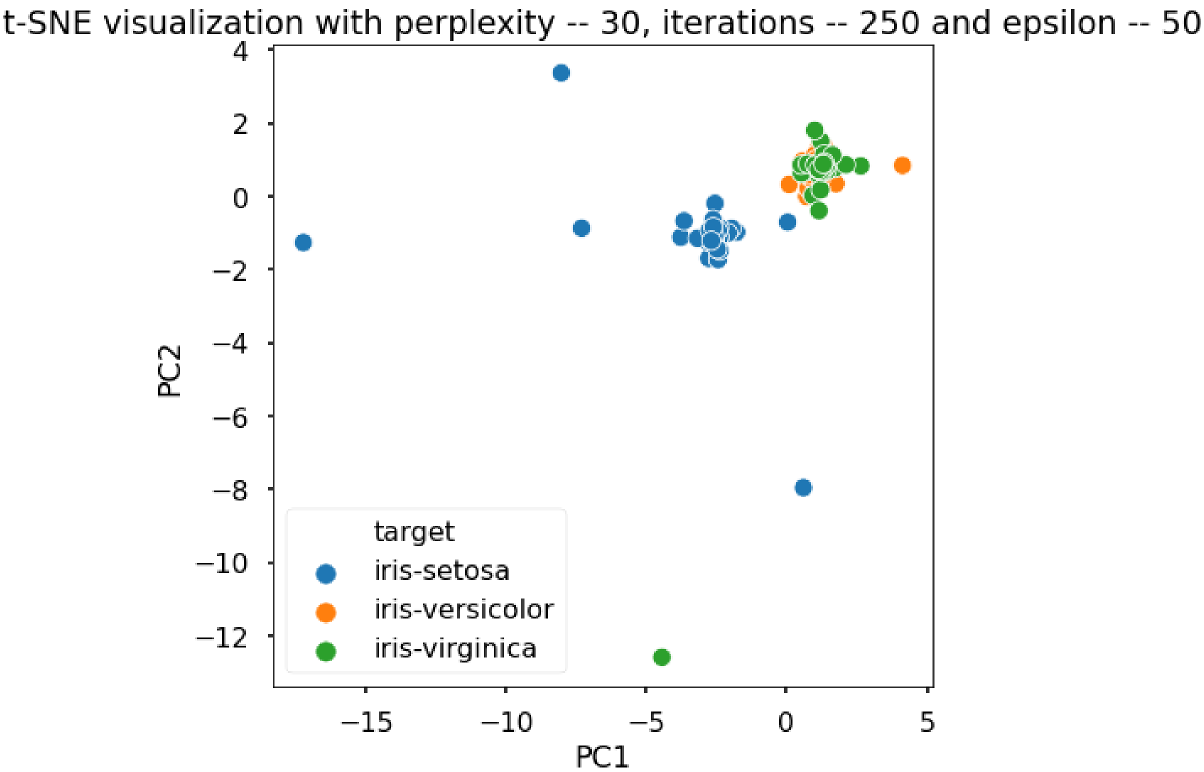

t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 10
PC2
PC1
30
20
10
0
−10
−3
−2
−1
0
1
2
3
target
iris-setosa
iris-versicolor
iris-virginica


t-SNE visualization with perplexity -- 30, iterations -- 2500 and epsilon -- 10
PC2
PC1
30
20
10
0
−10
−2
−1
0
1
2
3
4
target
iris-setosa
iris-versicolor
iris-virginica

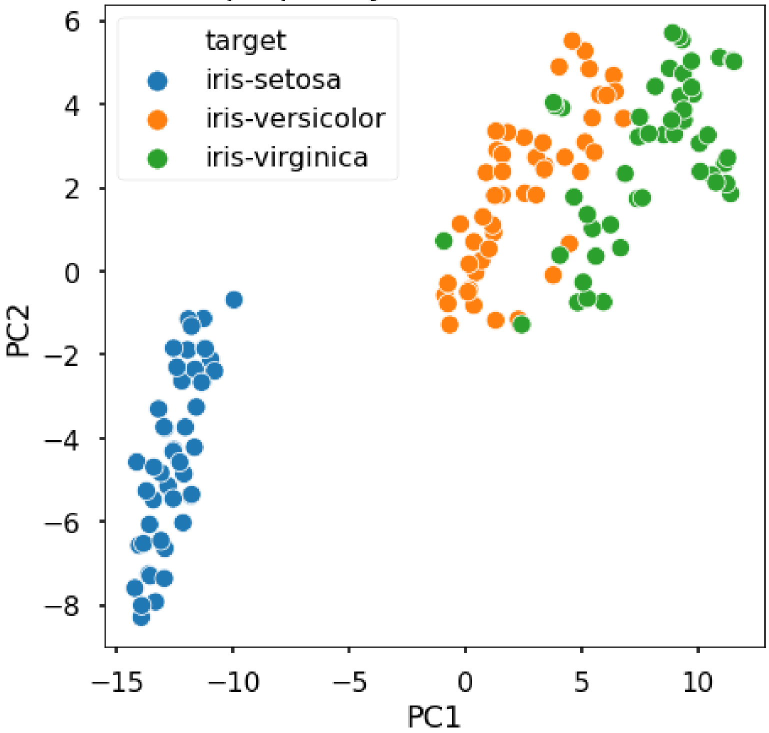

t-SNE visualization with perplexity -- 30, iterations -- 3500 and epsilon -- 10
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
30
20
10
0
−10
−2
−1
0
1
2
3
4


t-SNE visualization with perplexity -- 30, iterations -- 4000 and epsilon -- 10
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
30
20
10
0
−10
−2
−1
0
1
2
3
4

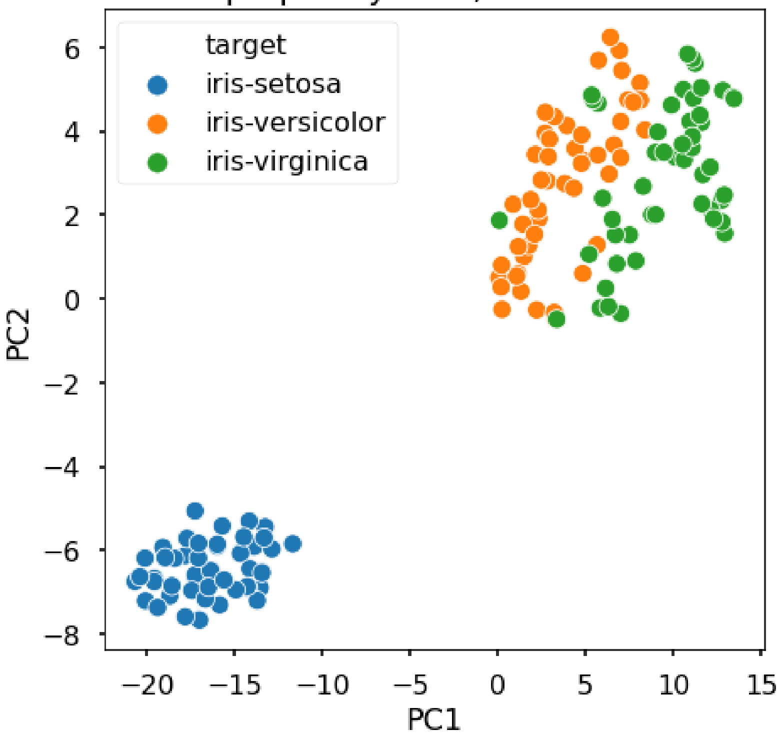

t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 10
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
30
20
10
0
−10
−2
−1
0
1
2
3
4

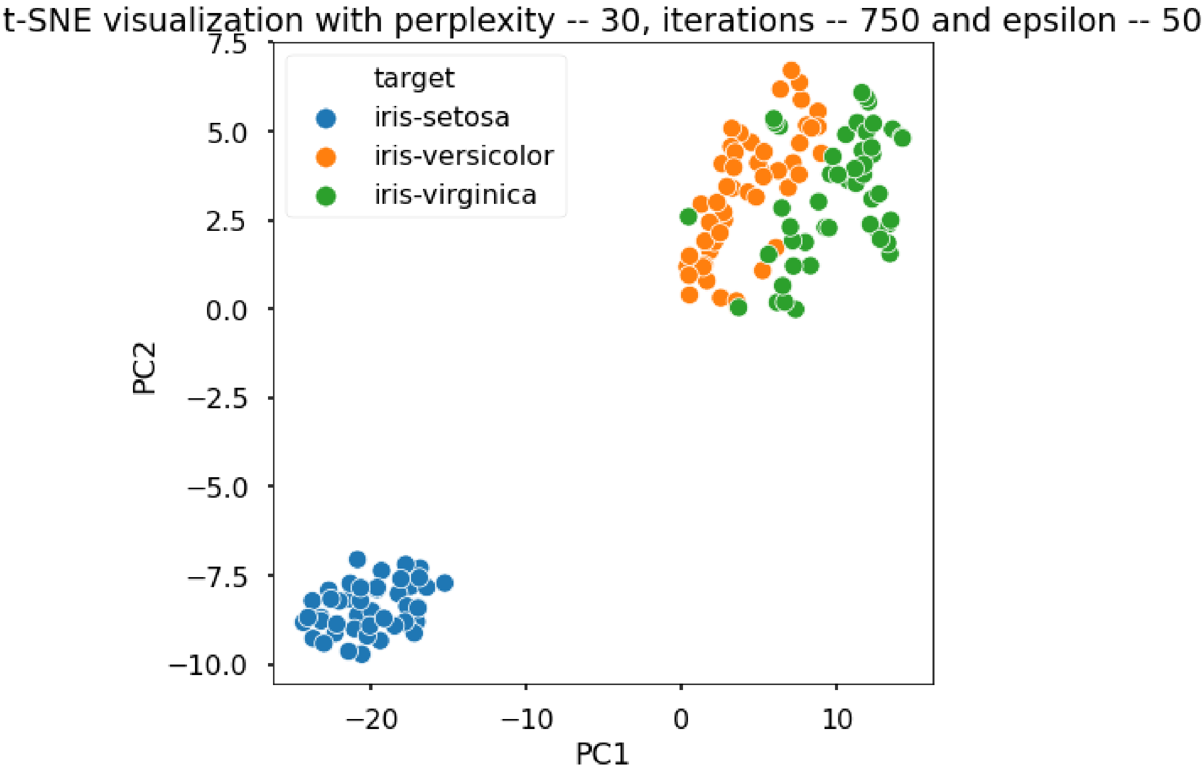

t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 50
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
4
2
0
−2
−4
−6
−8
−10
−12
−15
−10
−5
0
5

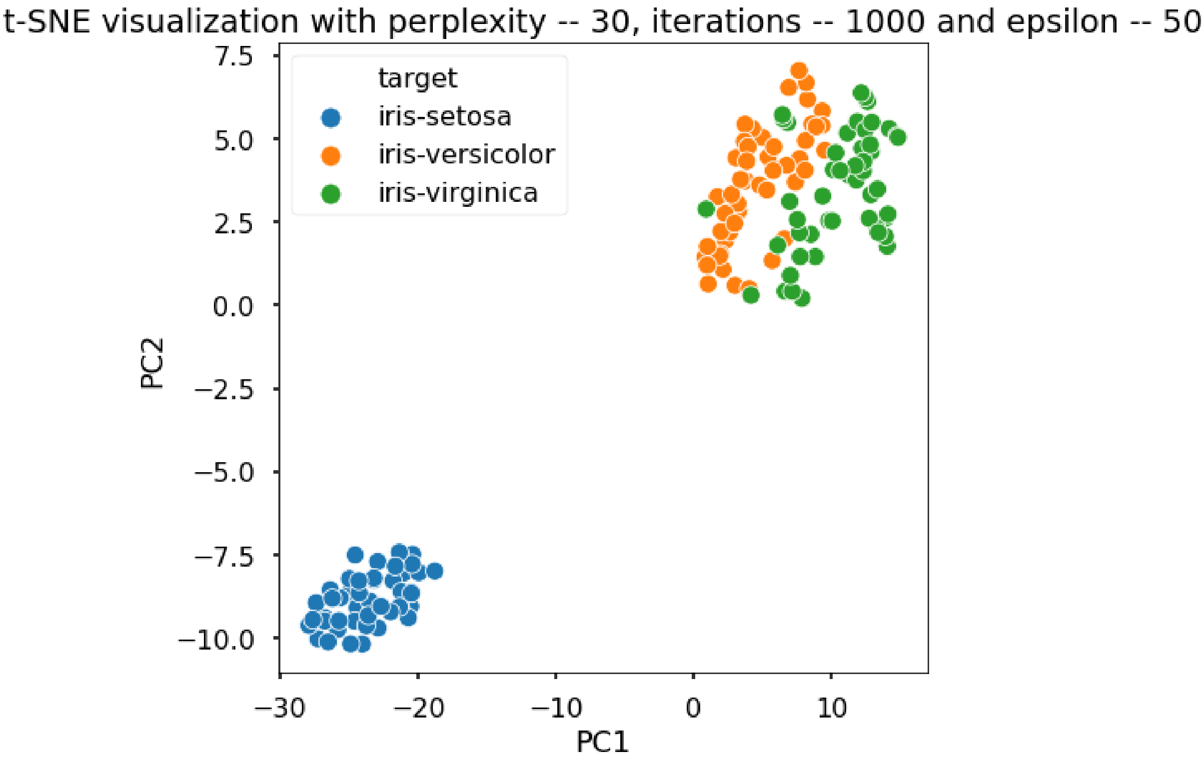
t-SNE visualization with perplexity -- 30, iterations -- 350 and epsilon -- 50



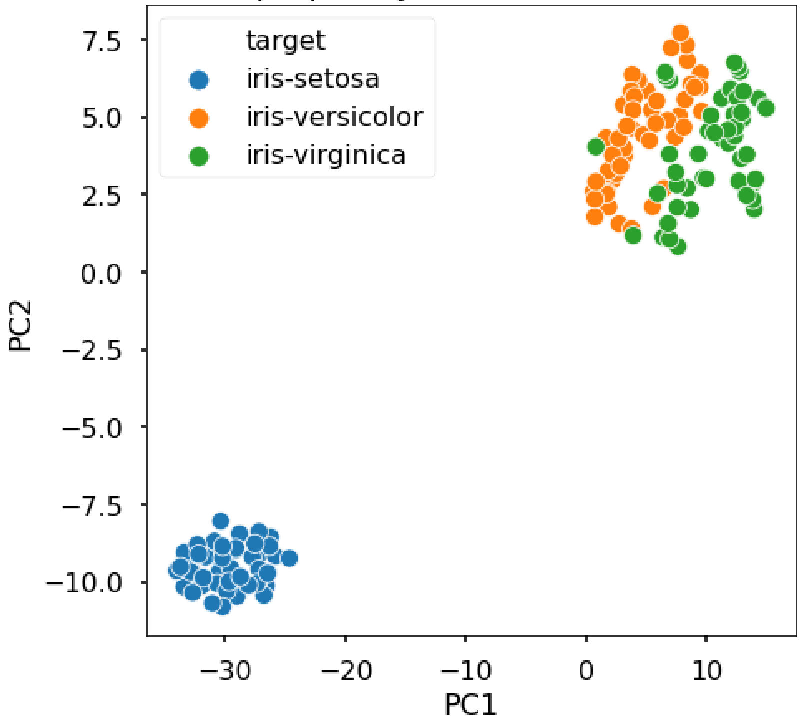
t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 50



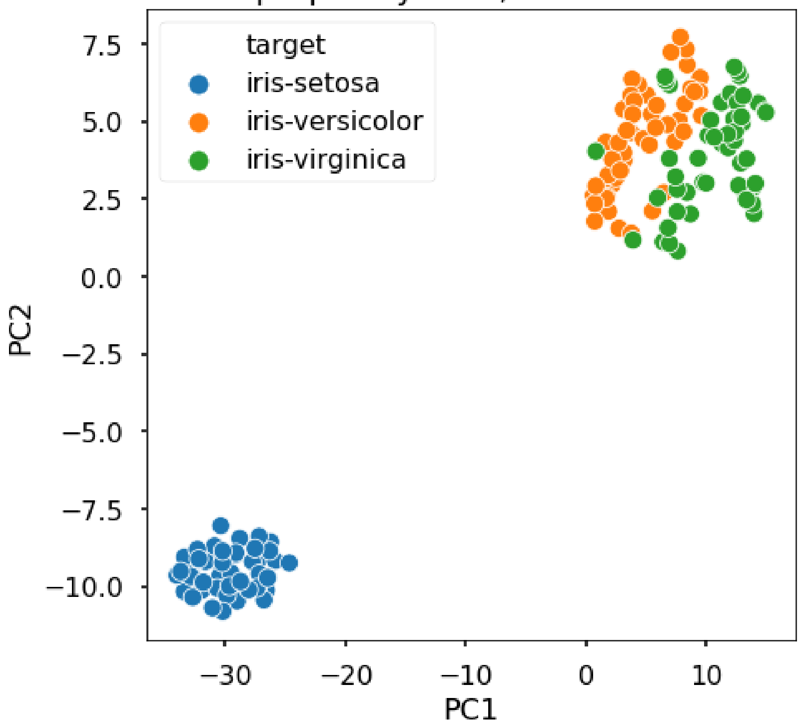

t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 50
target
iris-setosa
iris-versicolor
iris-virginica
PC2
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−20
−10
0
10
PC1


t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 50
target
iris-setosa
iris-versicolor
iris-virginica
PC2
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−30
−20
−10
0
10
PC1

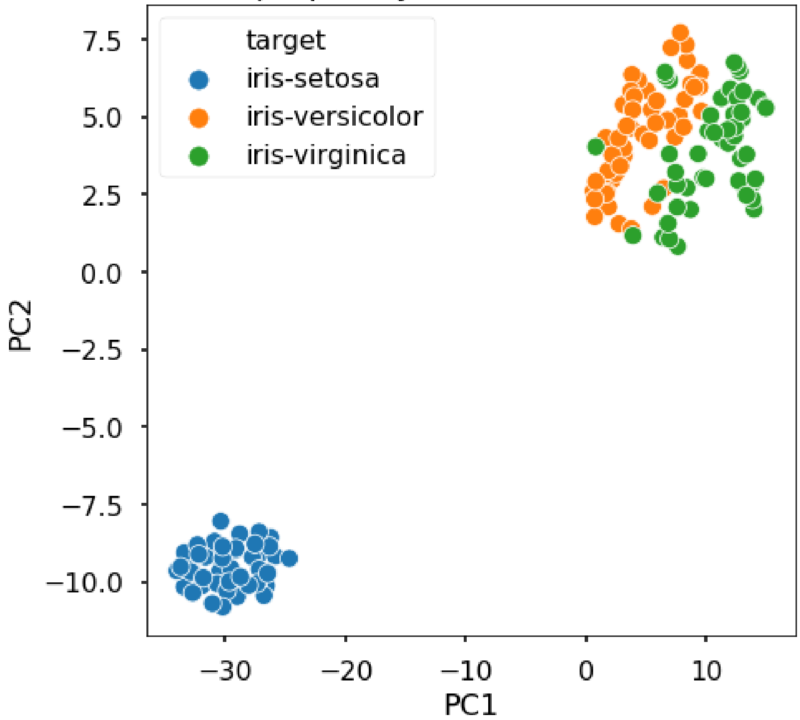
t-SNE visualization with perplexity -- 30, iterations -- 2500 and epsilon -- 50



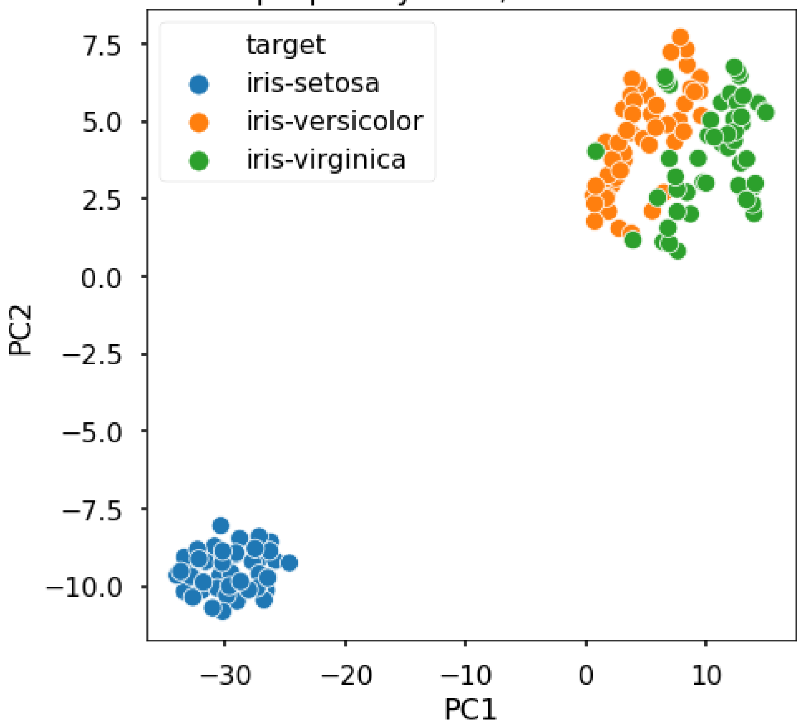
t-SNE visualization with perplexity -- 30, iterations -- 3500 and epsilon -- 50

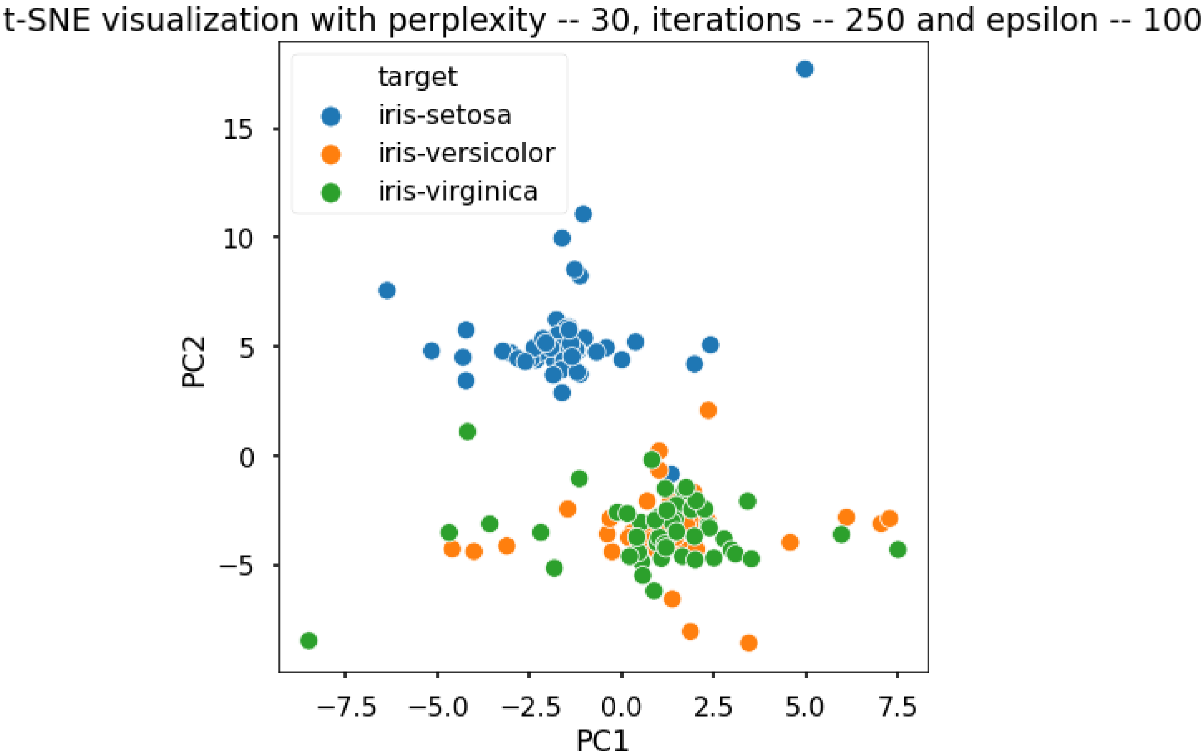


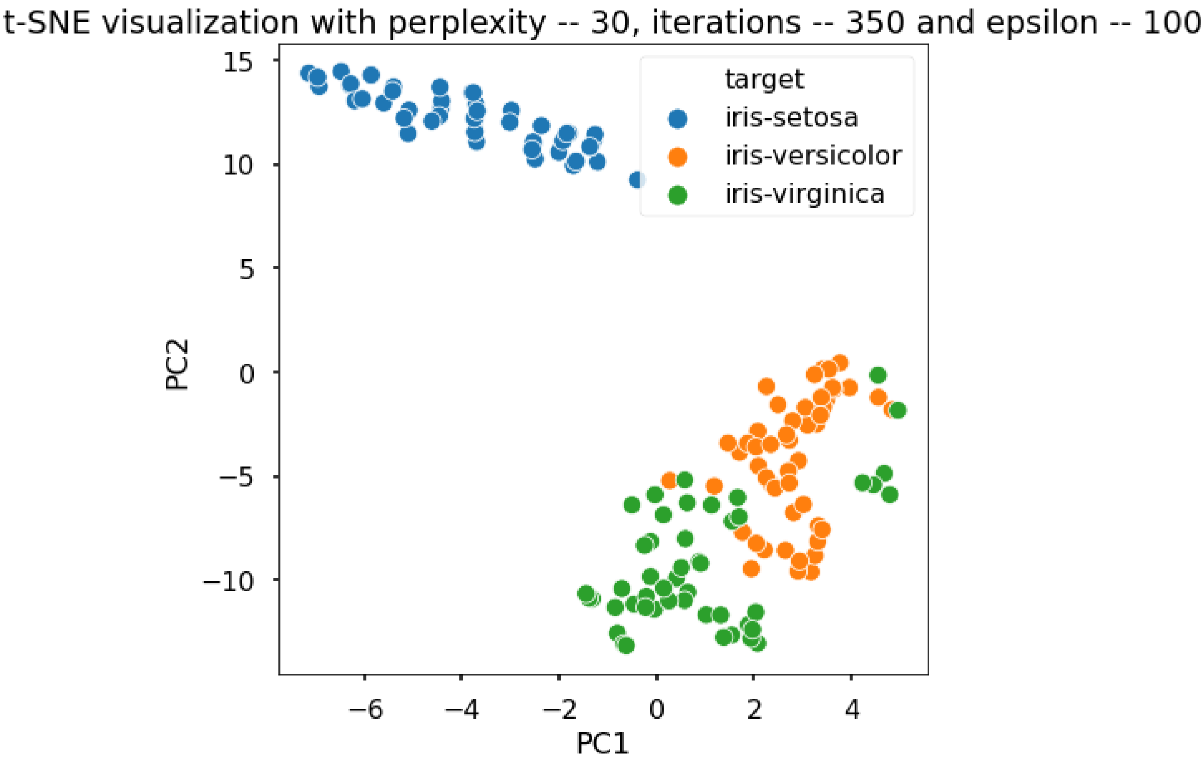
t-SNE visualization with perplexity -- 30, iterations -- 4000 and epsilon -- 50

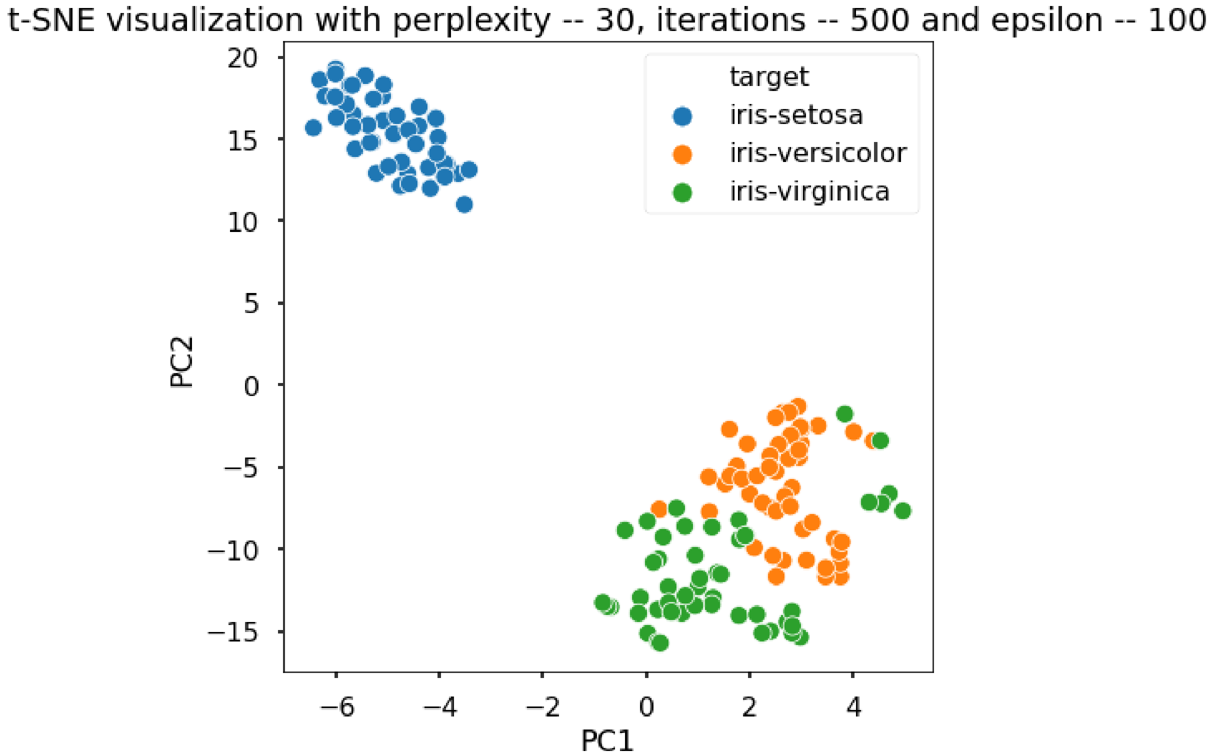


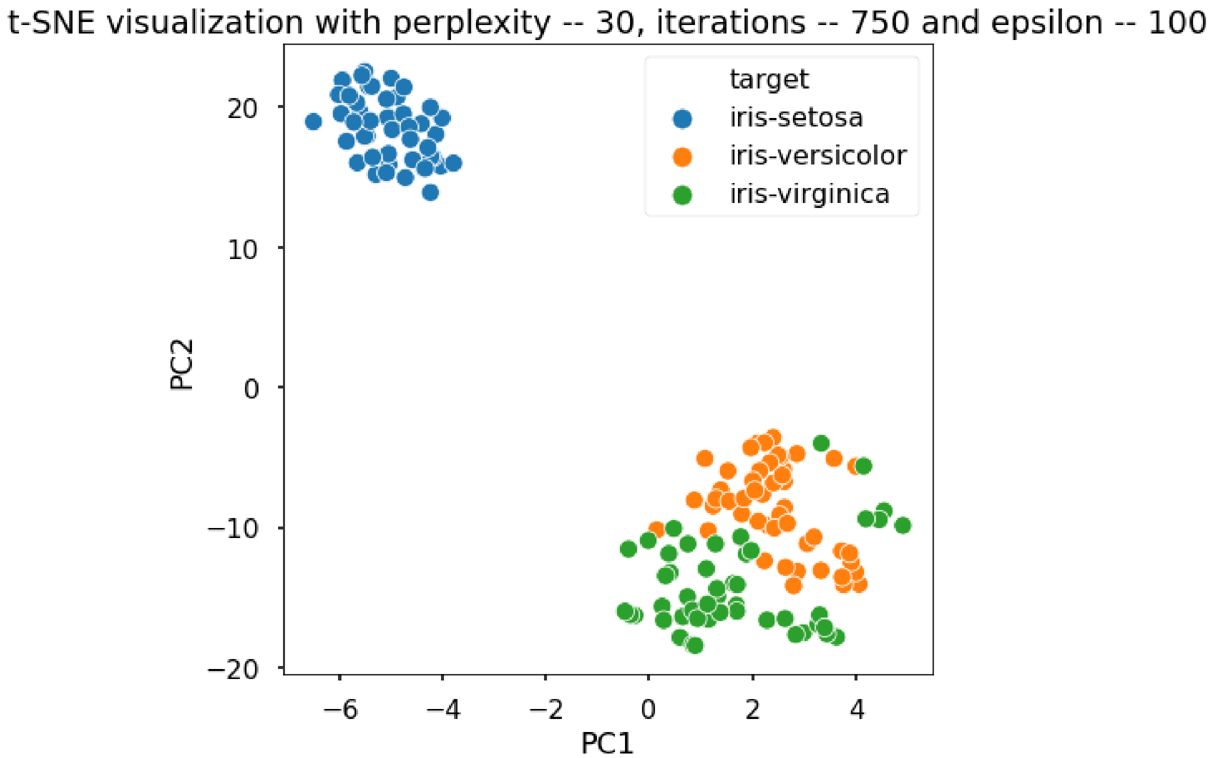
t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 50

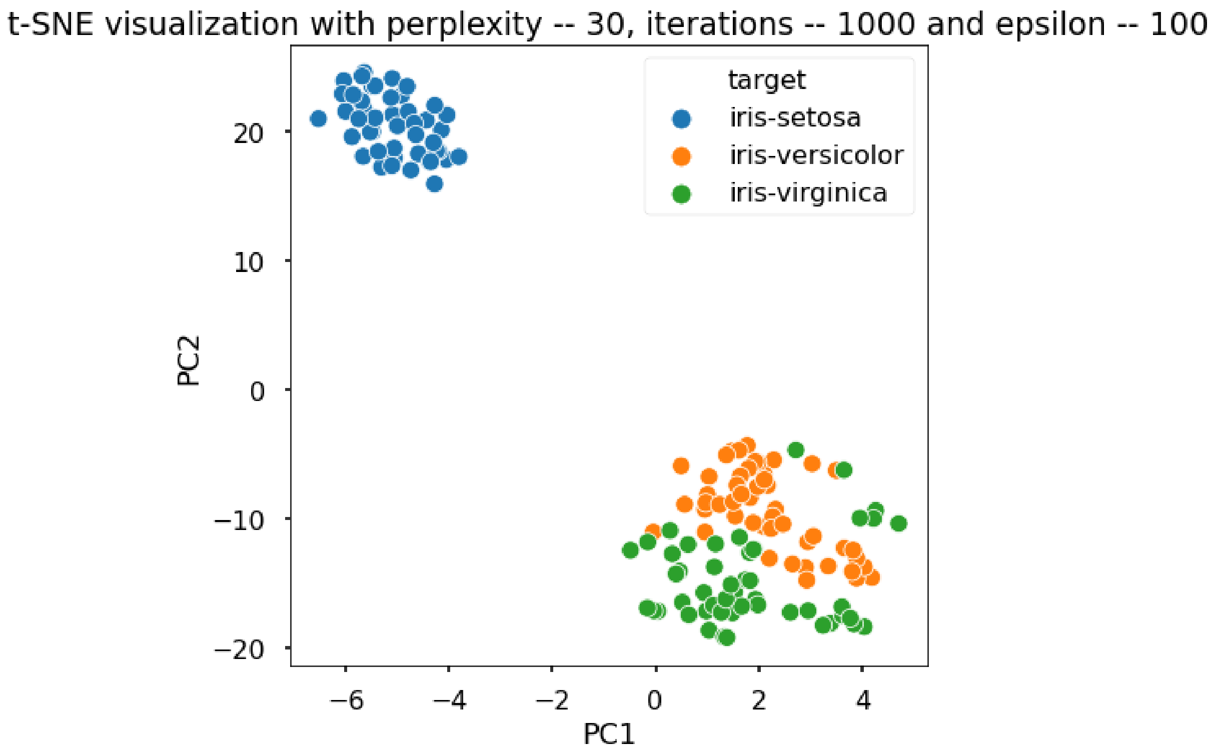


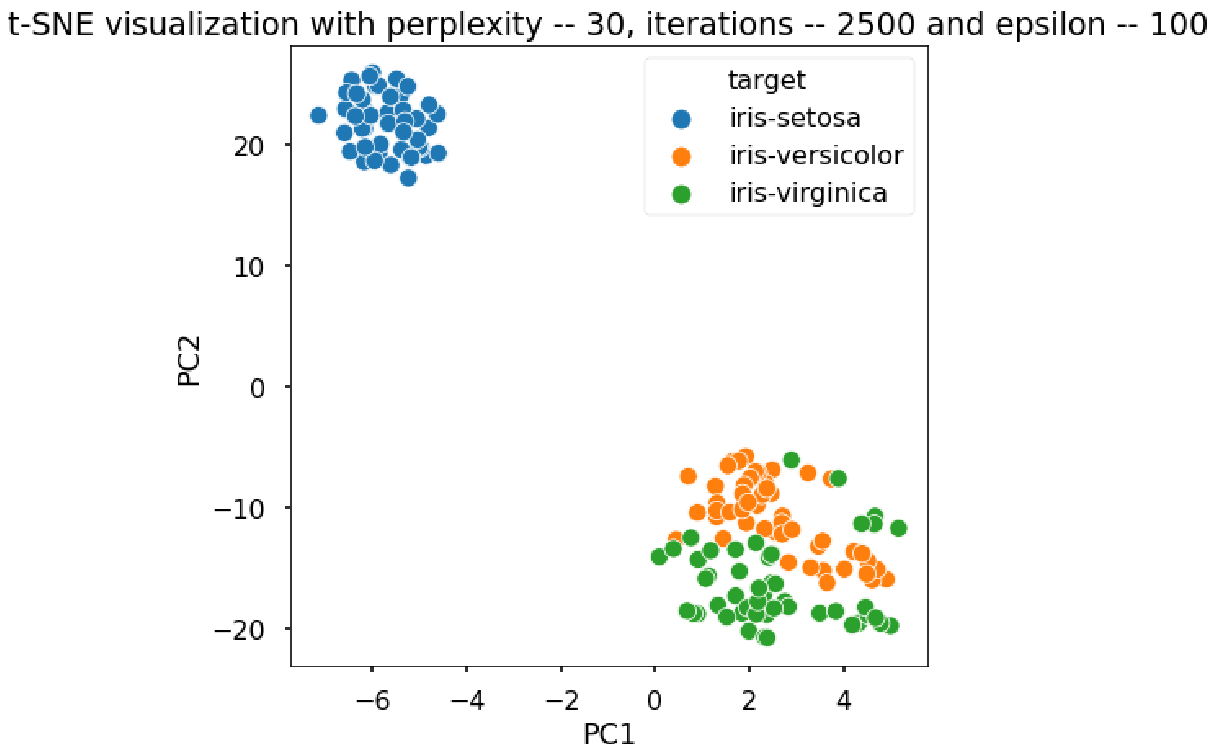

t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 100
target
iris-setosa
iris-versicolor
iris-virginica
15
10
5
0
−5
PC2
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
PC1

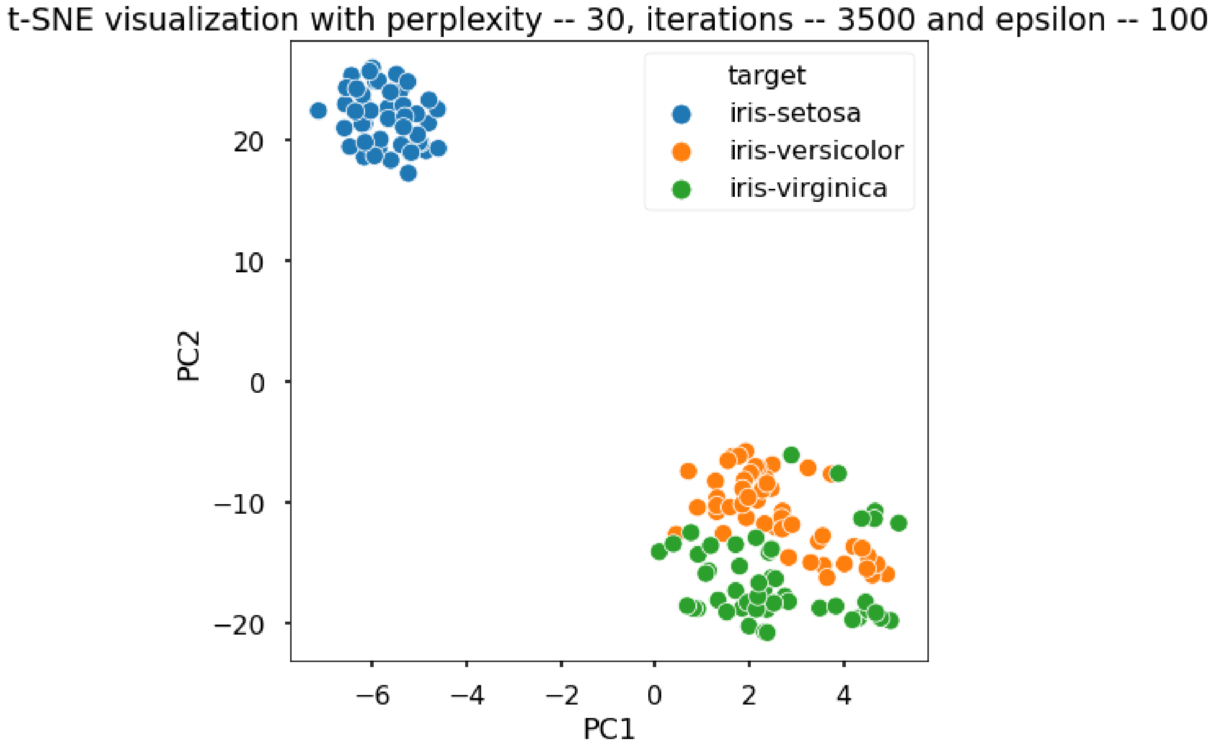

t-SNE visualization with perplexity -- 30, iterations -- 350 and epsilon -- 100
target
iris-setosa
iris-versicolor
iris-virginica
15
10
5
0
−5
−10
PC2
−6
−4
−2
0
2
4
PC1

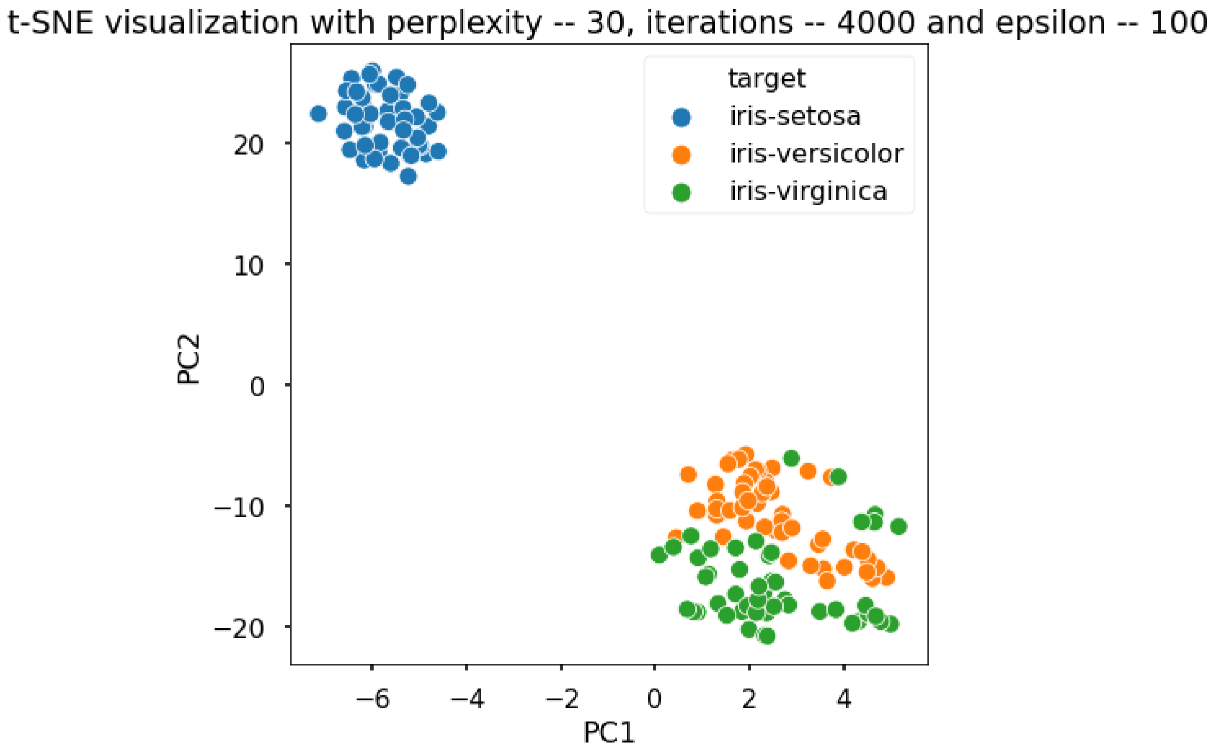

t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 100
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
20
15
10
5
0
−5
−10
−15
−6
−4
−2
0
2
4

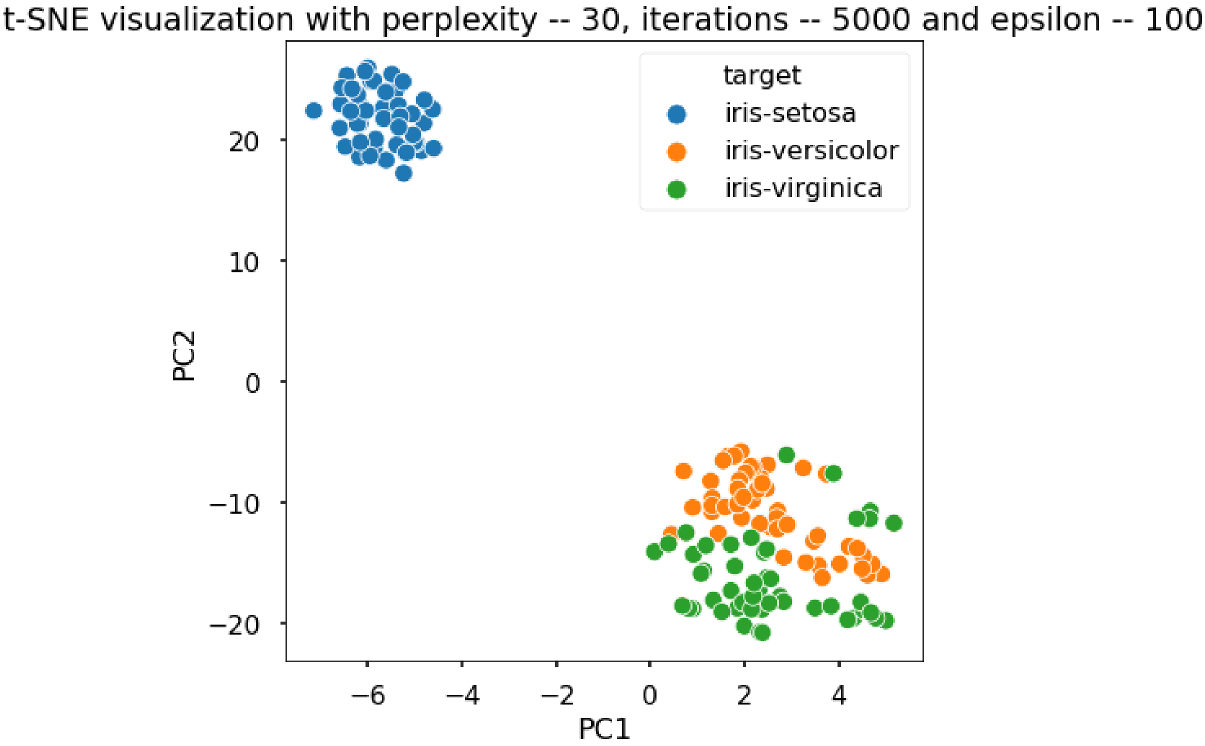

t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 100
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
20
10
0
−10
−20
−6
−4
−2
0
2
4

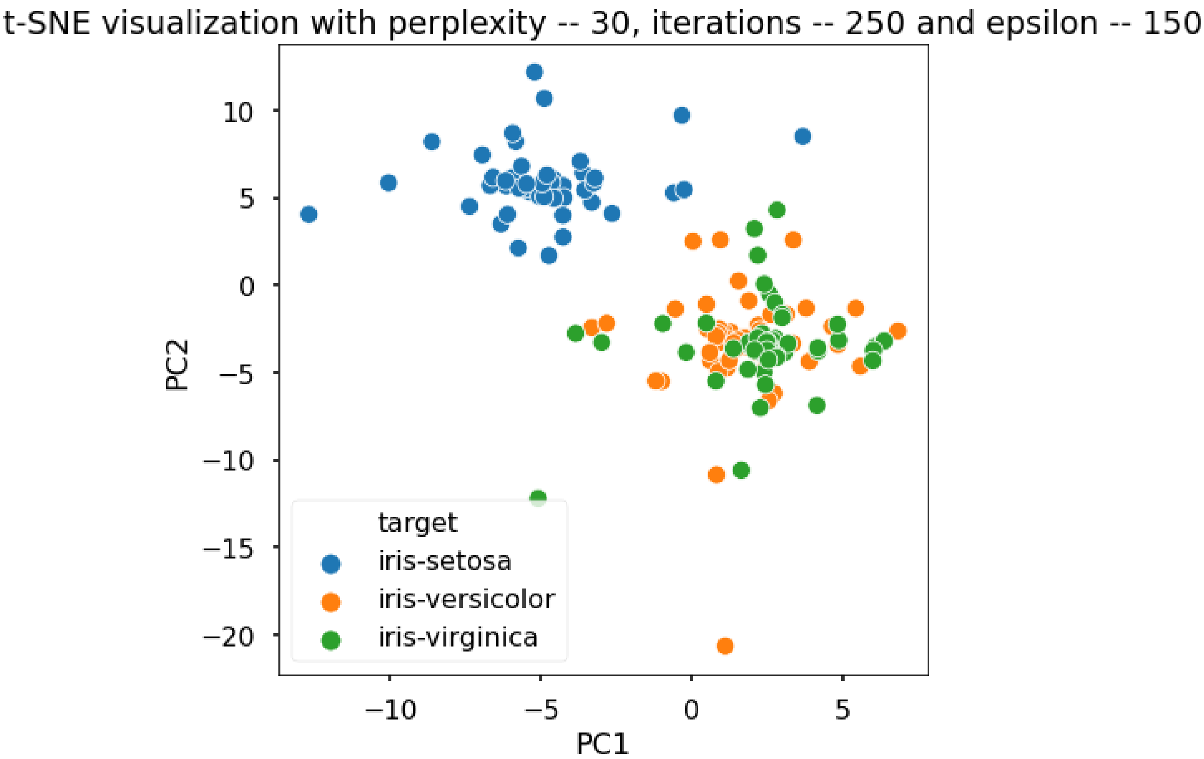

t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 100
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
20
10
0
−10
−20
−6
−4
−2
0
2
4

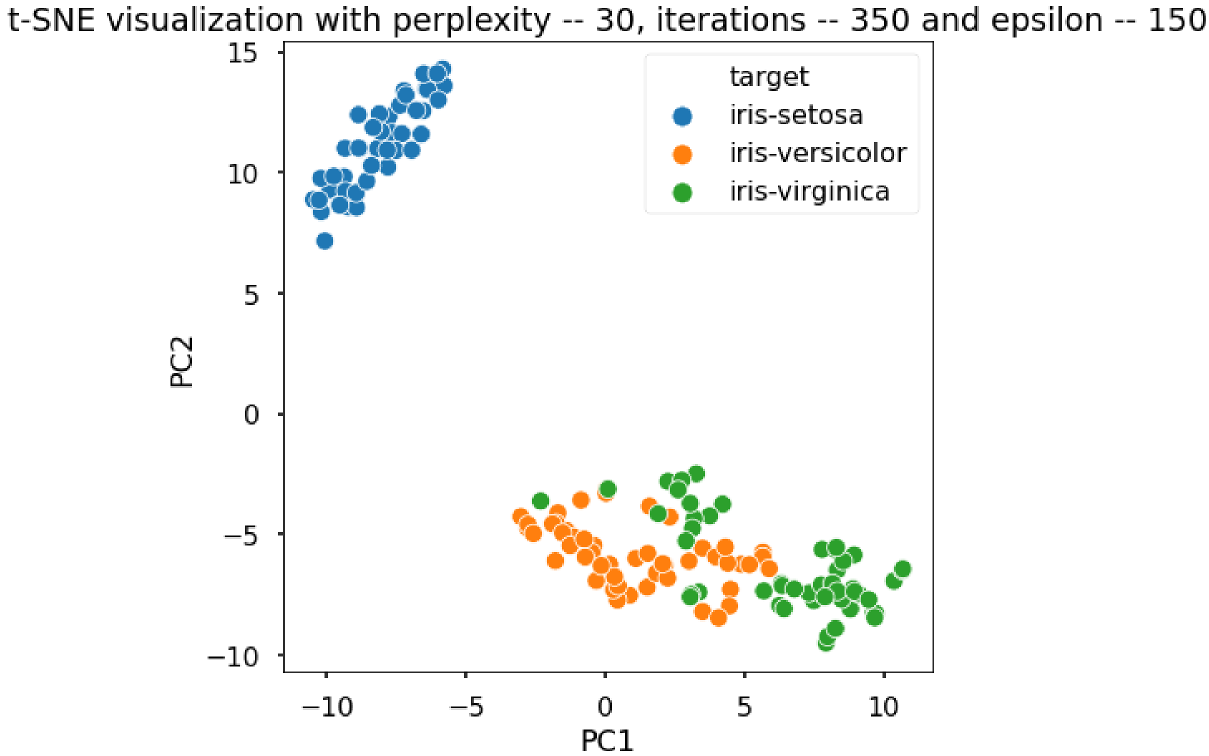

t-SNE visualization with perplexity -- 30, iterations -- 2500 and epsilon -- 100
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
20
10
0
−10
−20
−6
−4
−2
0
2
4

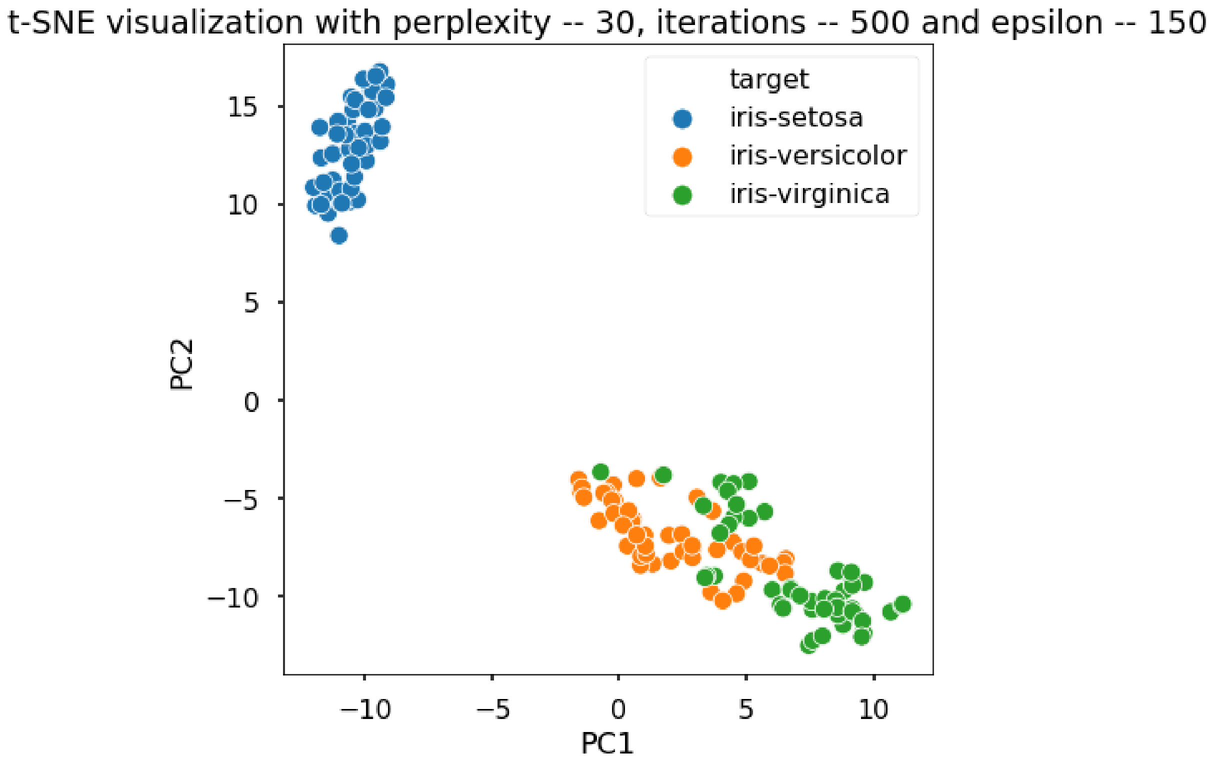

t-SNE visualization with perplexity -- 30, iterations -- 3500 and epsilon -- 100
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
20
10
0
−10
−20
−6
−4
−2
0
2
4


t-SNE visualization with perplexity -- 30, iterations -- 4000 and epsilon -- 100
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
20
10
0
−10
−20
−6
−4
−2
0
2
4

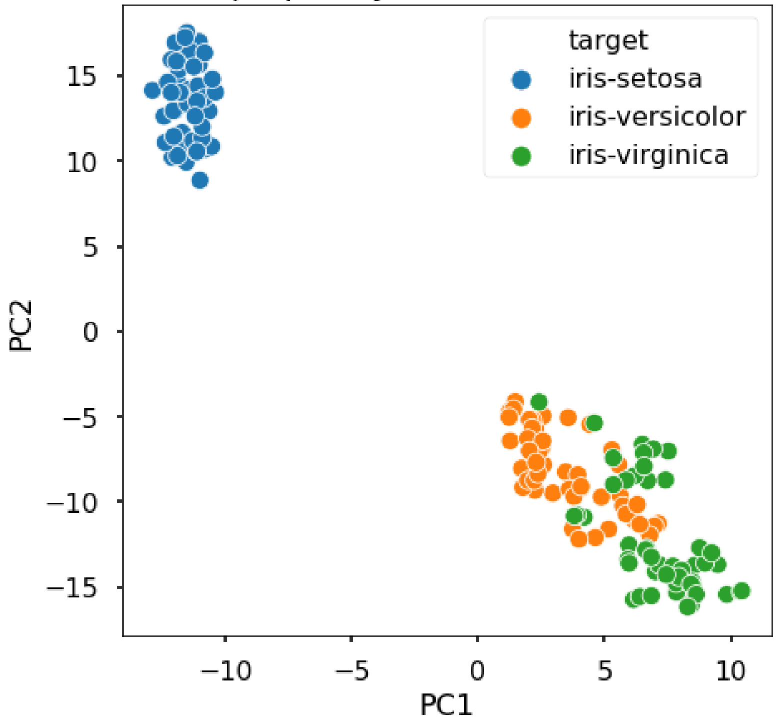

t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 100
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
20
10
0
−10
−20
−6
−4
−2
0
2
4


t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 150
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
10
5
0
−5
−10
−15
−20
−10
−5
0
5

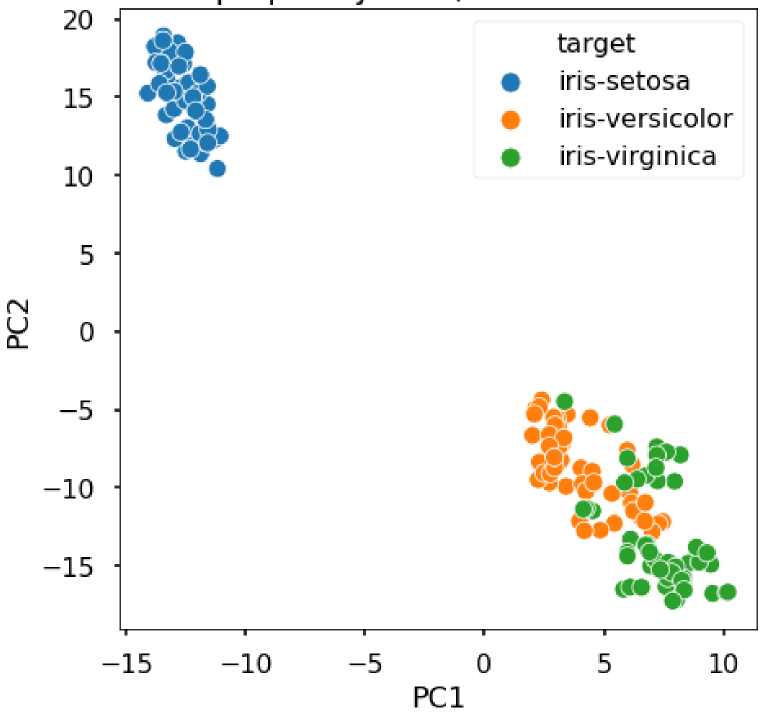

t-SNE visualization with perplexity -- 30, iterations -- 350 and epsilon -- 150
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
15
10
5
0
−5
−10
−10
−5
0
5
10


t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 150
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
15
10
5
0
−5
−10
−10
−5
0
5
10

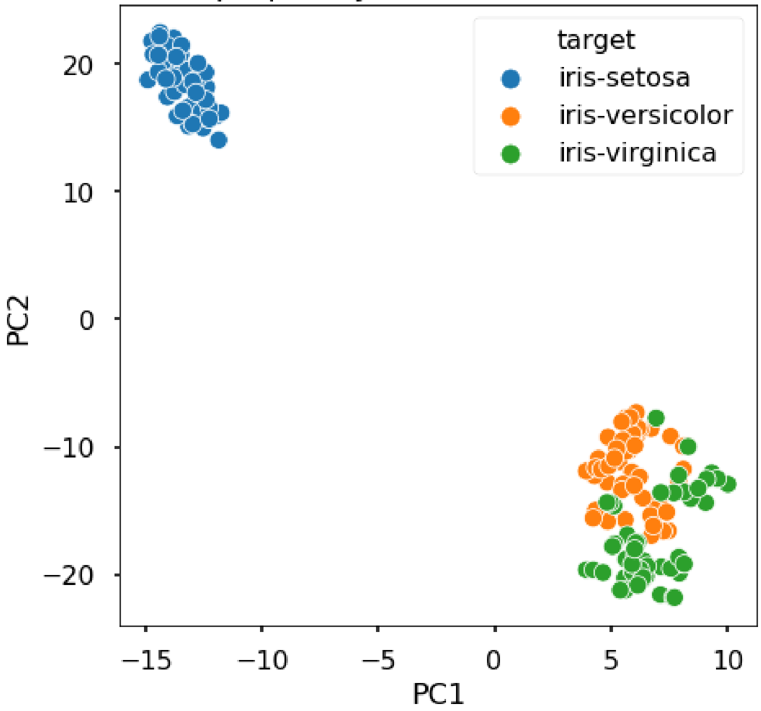
t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 150



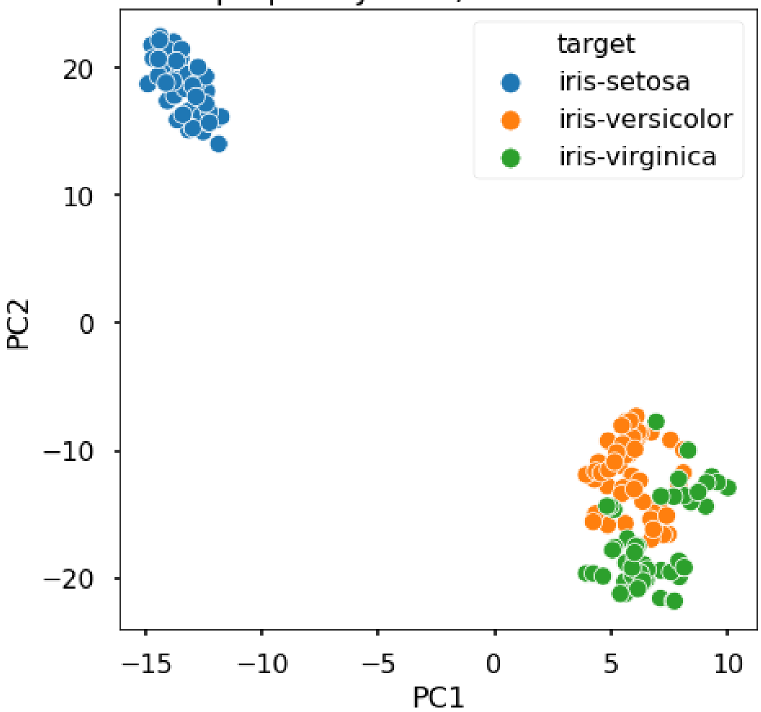
t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 150



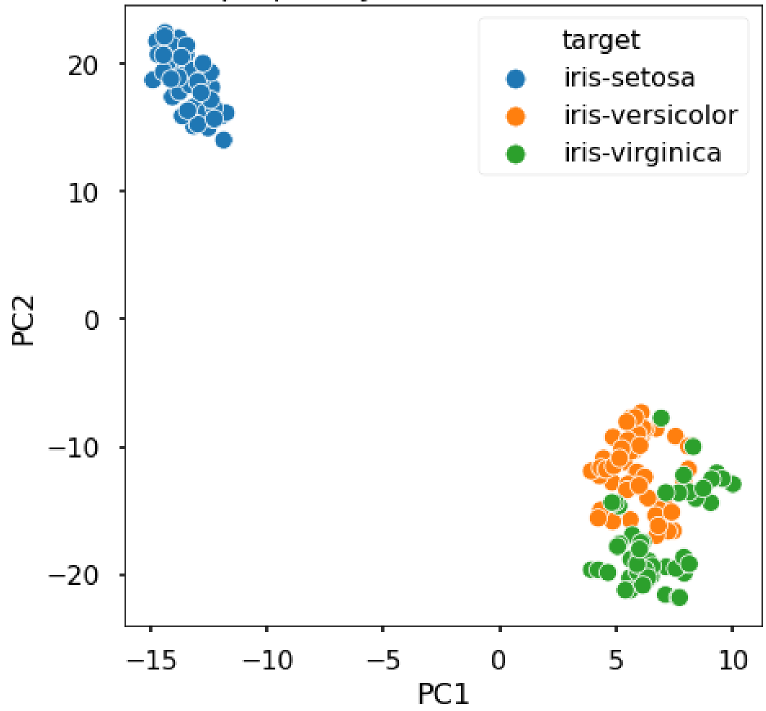
t-SNE visualization with perplexity -- 30, iterations -- 2500 and epsilon -- 150



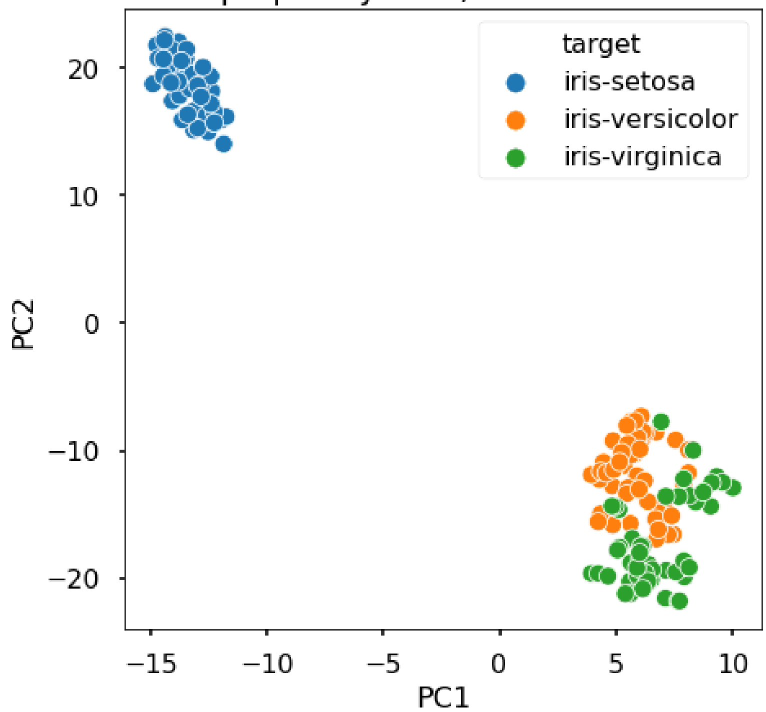
t-SNE visualization with perplexity -- 30, iterations -- 3500 and epsilon -- 150



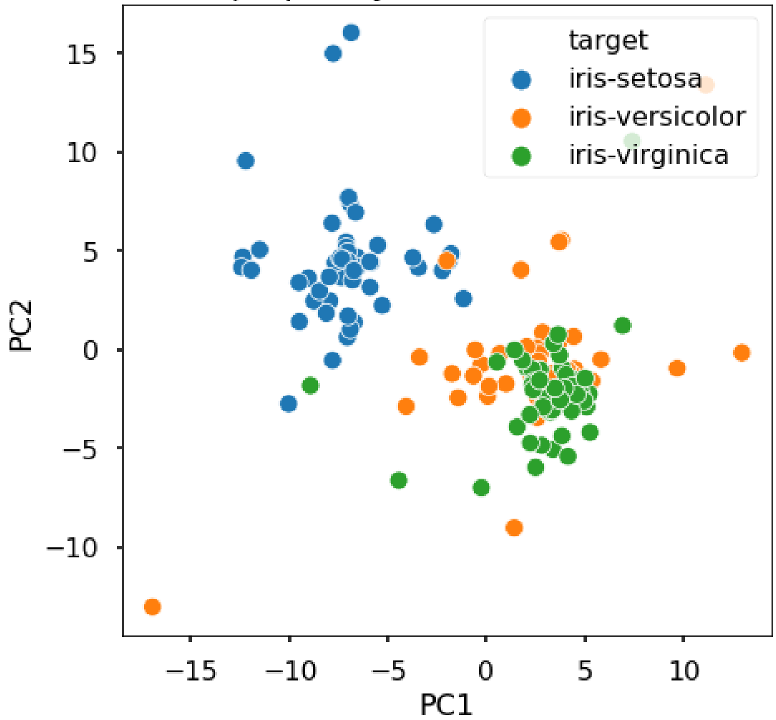
t-SNE visualization with perplexity -- 30, iterations -- 4000 and epsilon -- 150



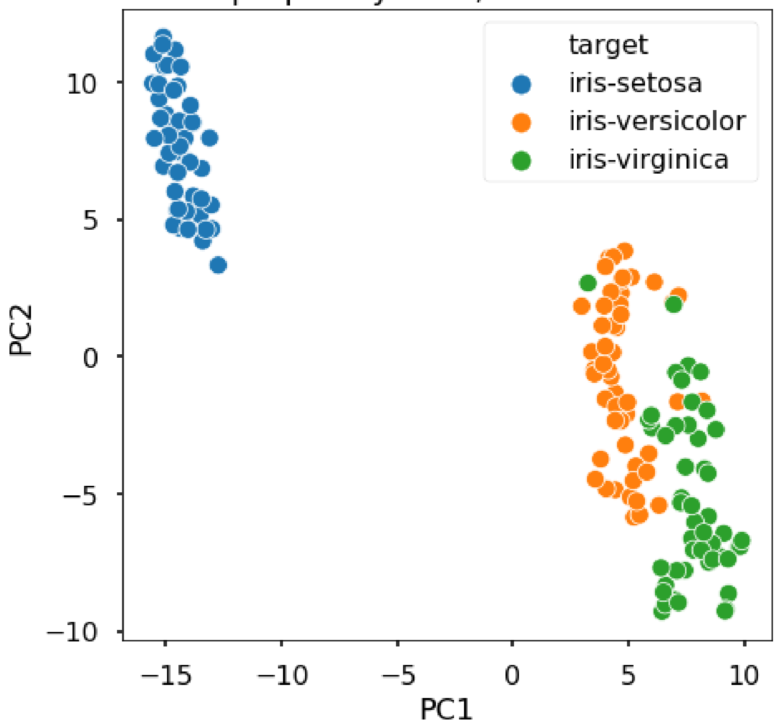
t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 150



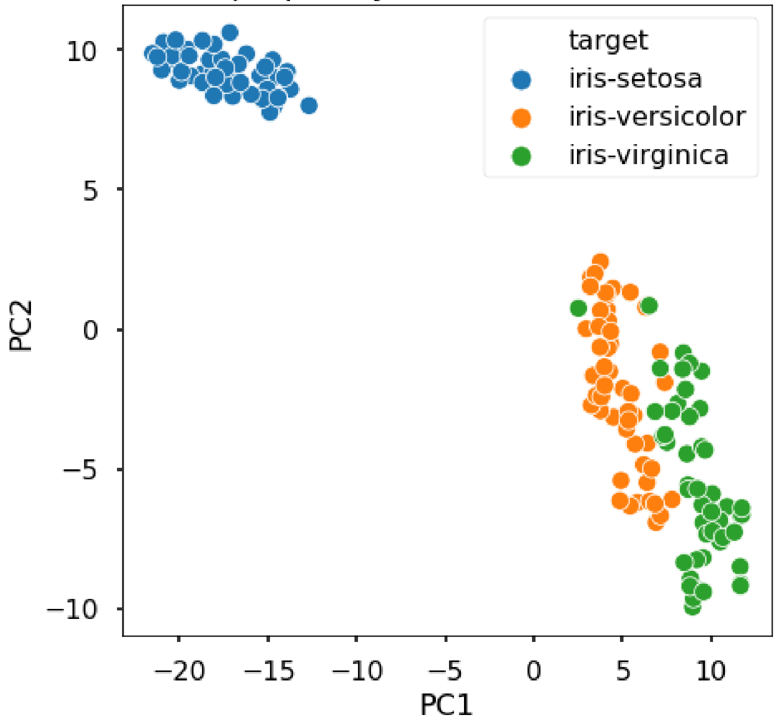
t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 200



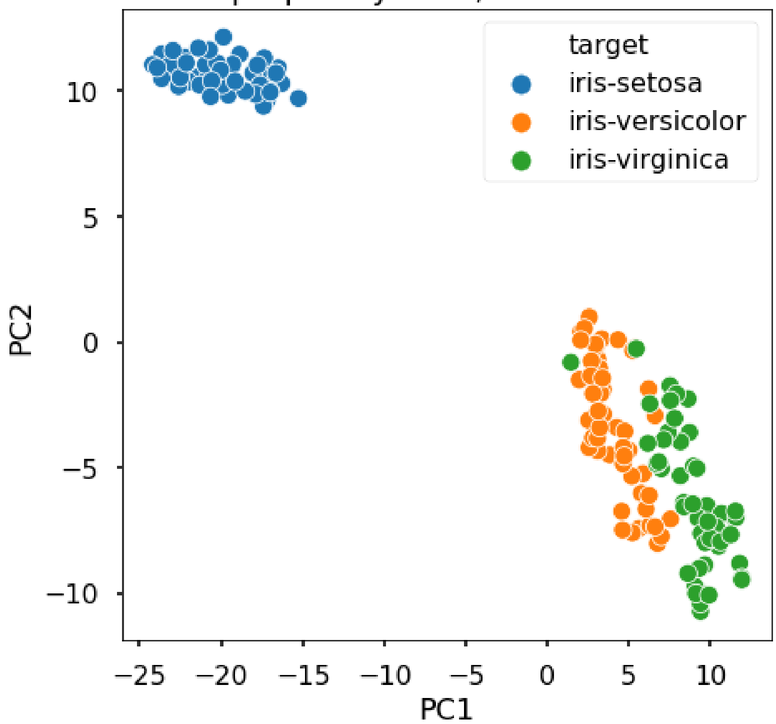
t-SNE visualization with perplexity -- 30, iterations -- 350 and epsilon -- 200



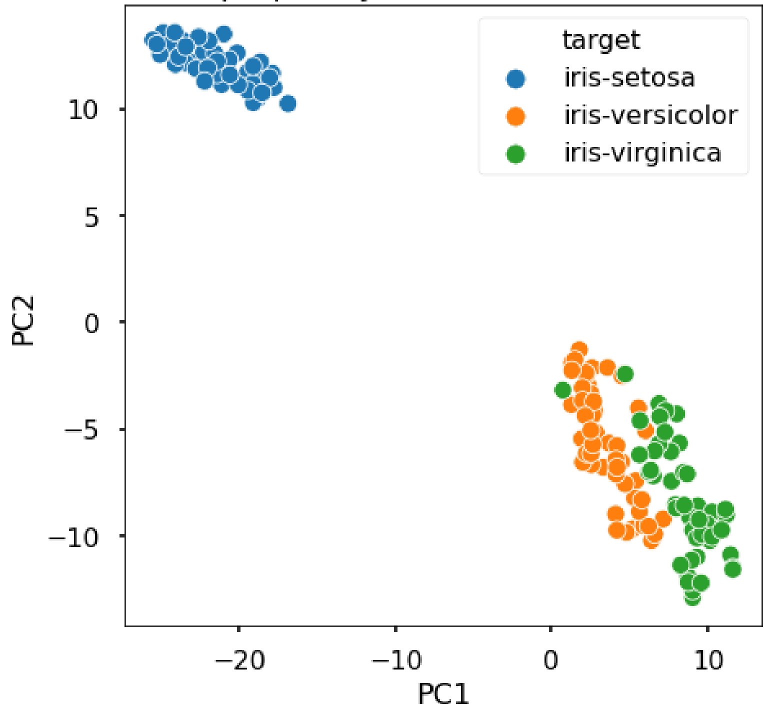
t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 200



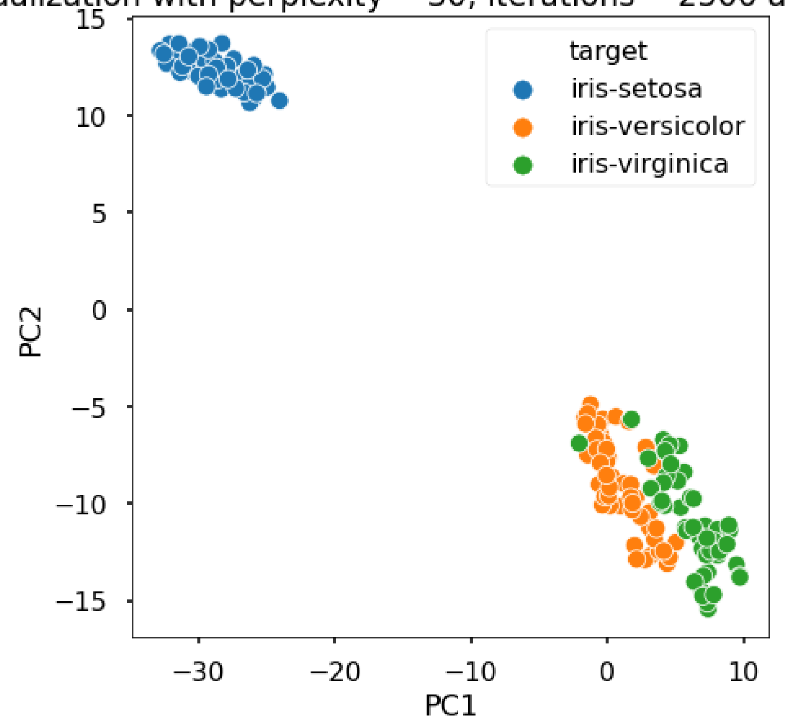
t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 200

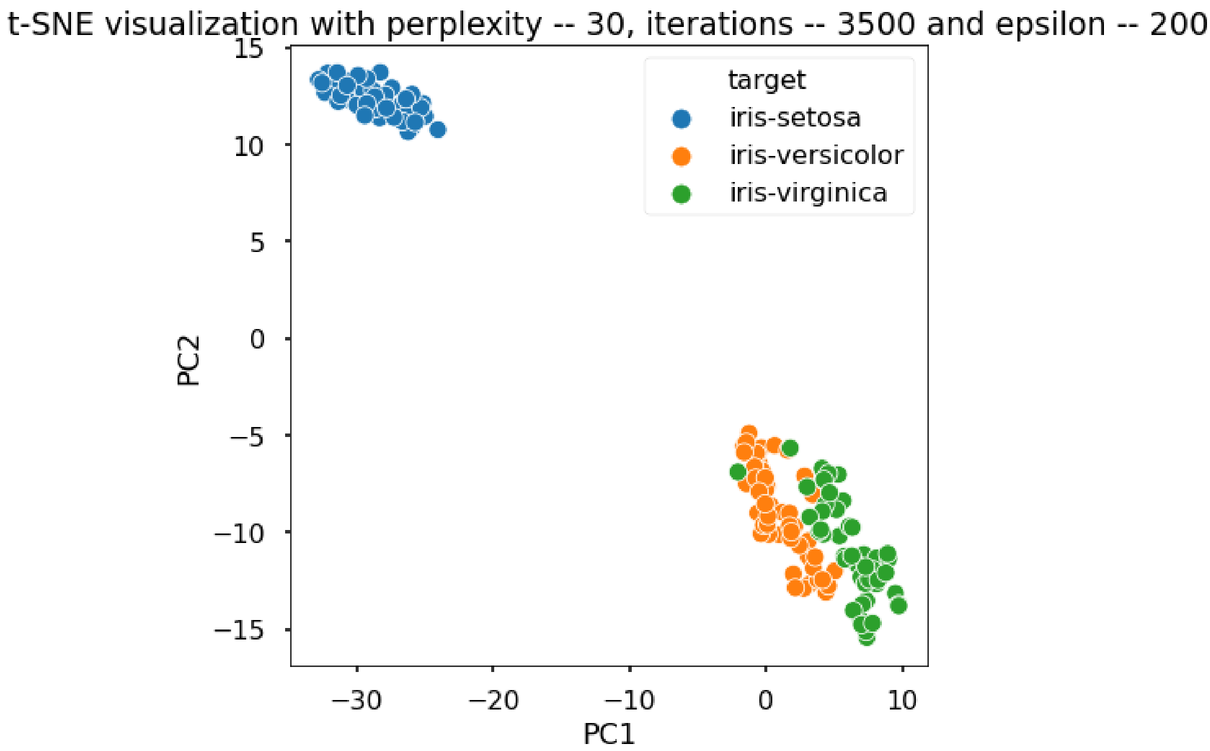


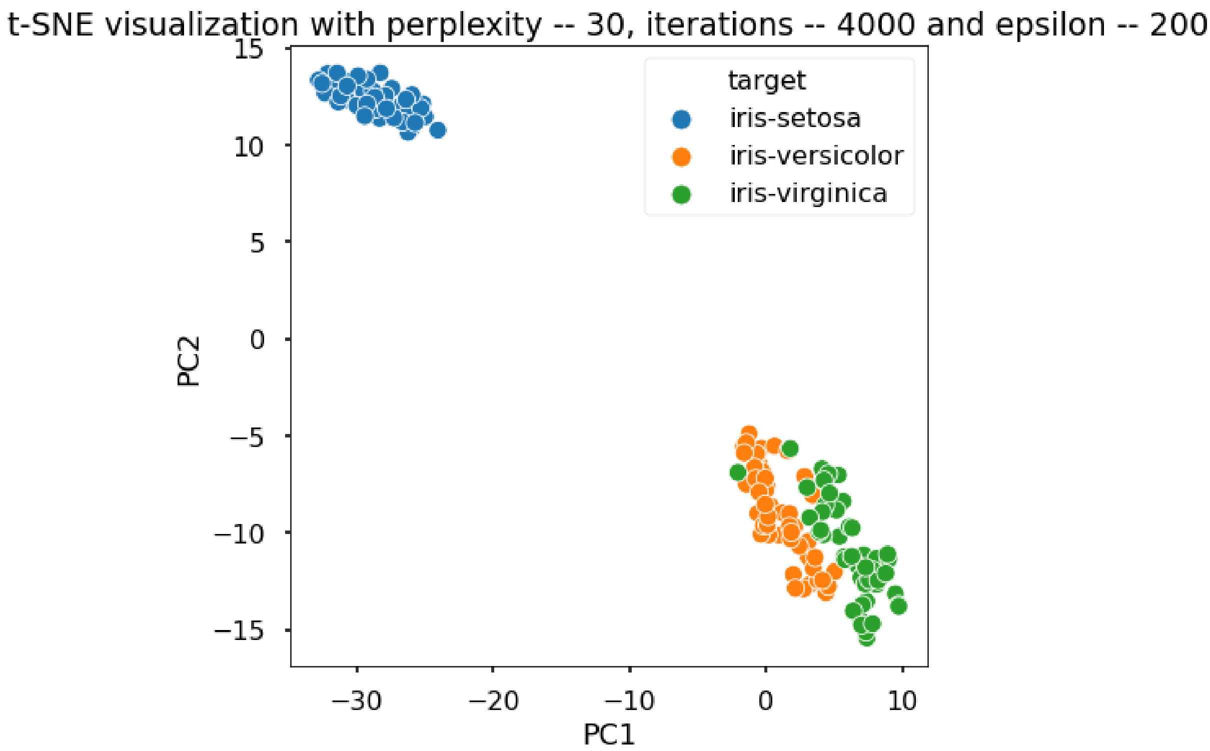
t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 200

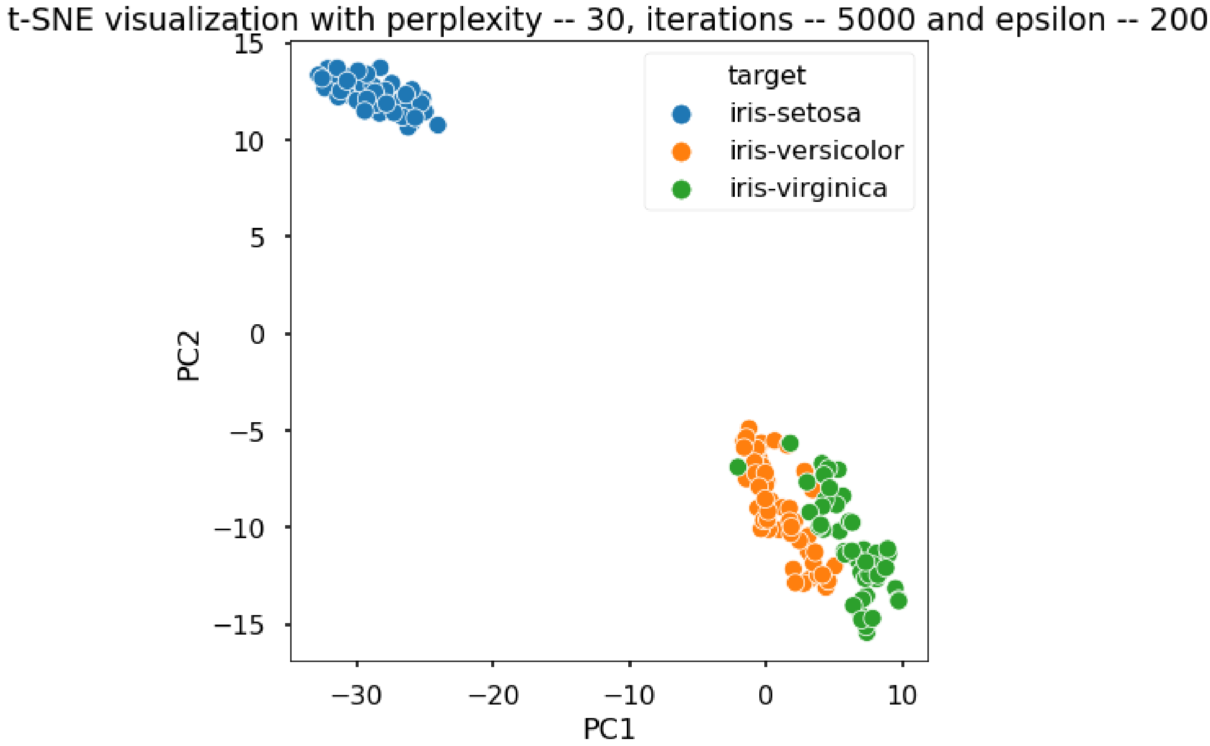


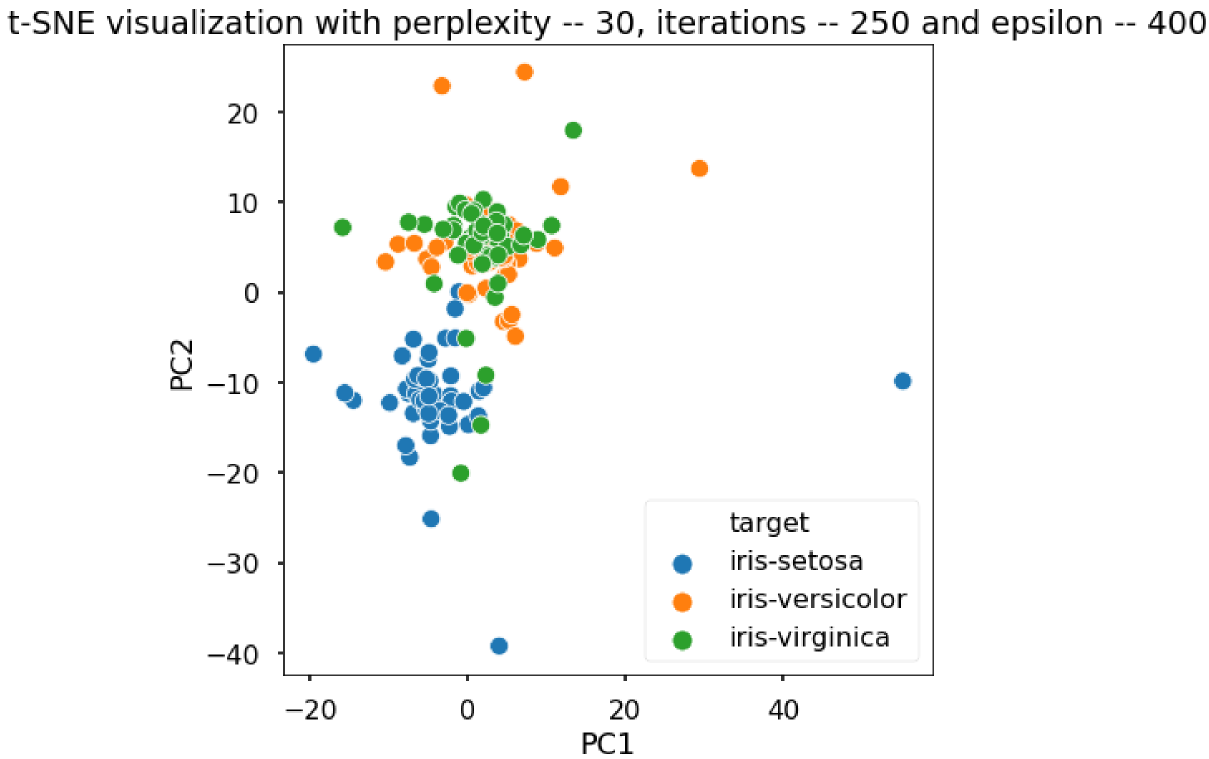
t-SNE visualization with perplexity -- 30, iterations -- 2500 and epsilon -- 200

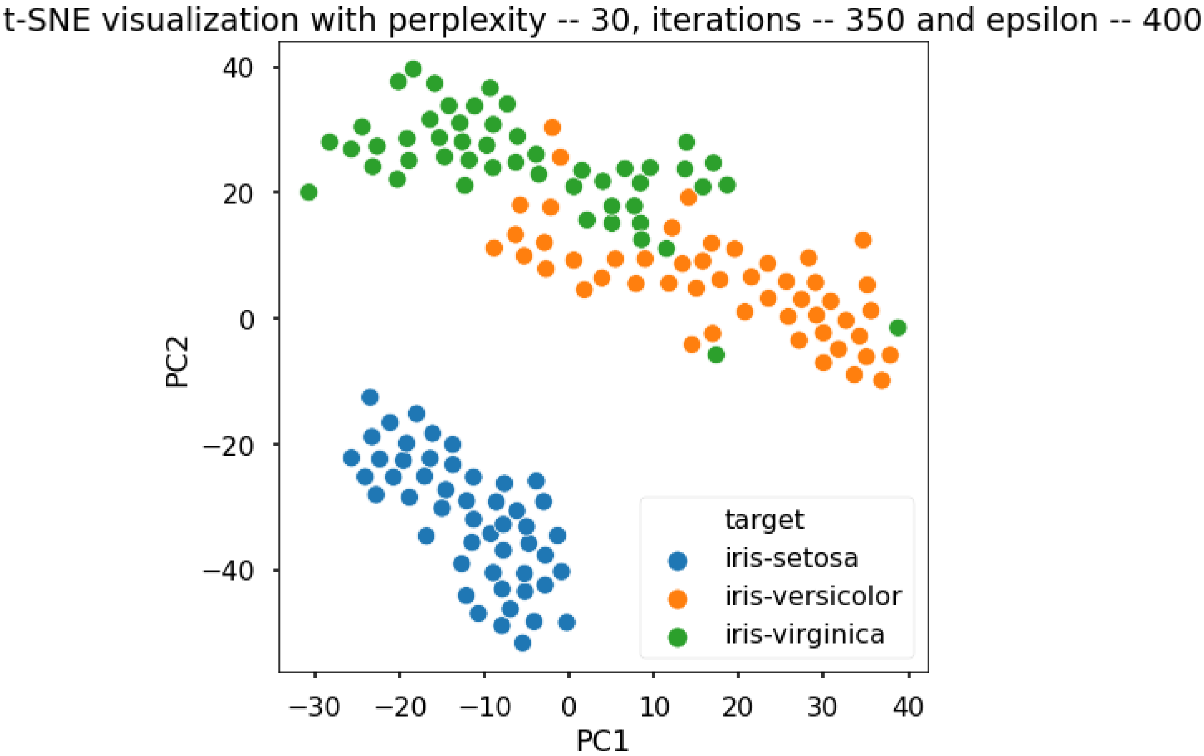


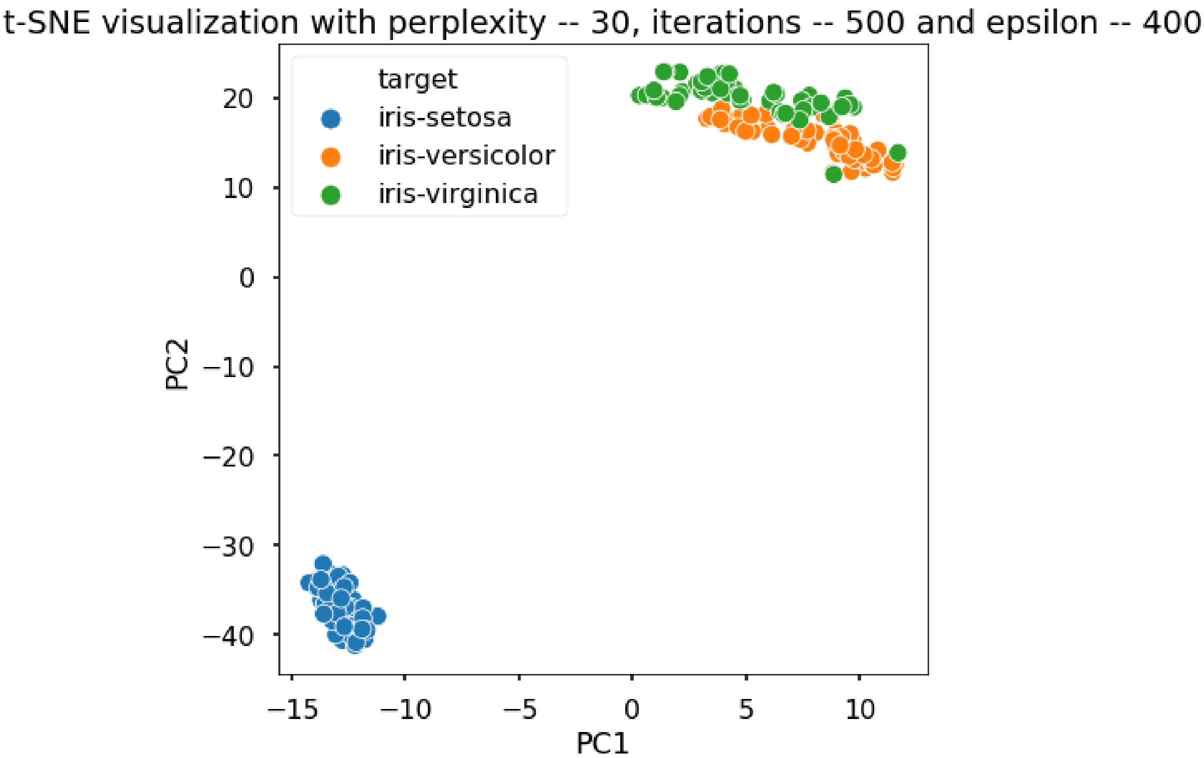

t-SNE visualization with perplexity -- 30, iterations -- 3500 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10


t-SNE visualization with perplexity -- 30, iterations -- 4000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10

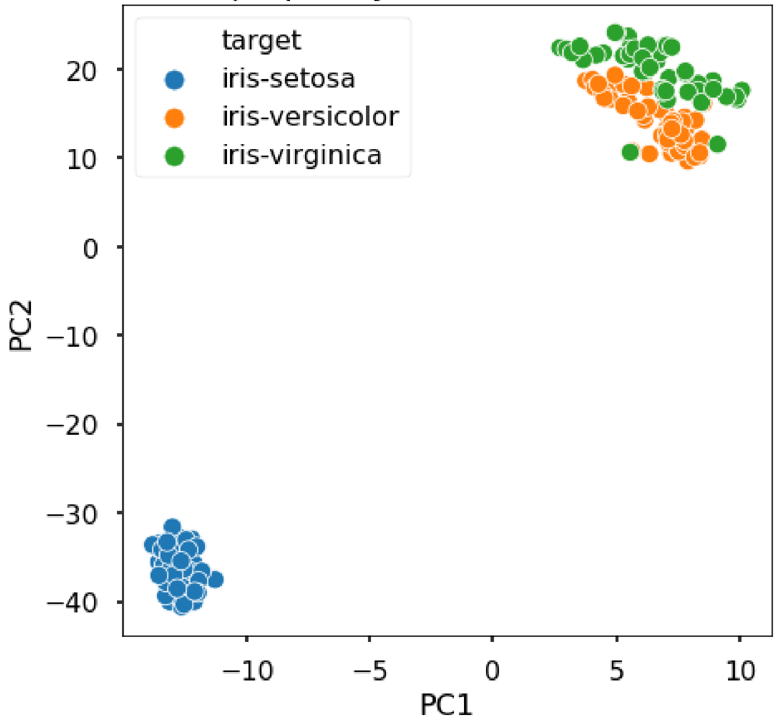

t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10


t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 400
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
20
10
0
−10
−20
−30
−40
−20
0
20
40

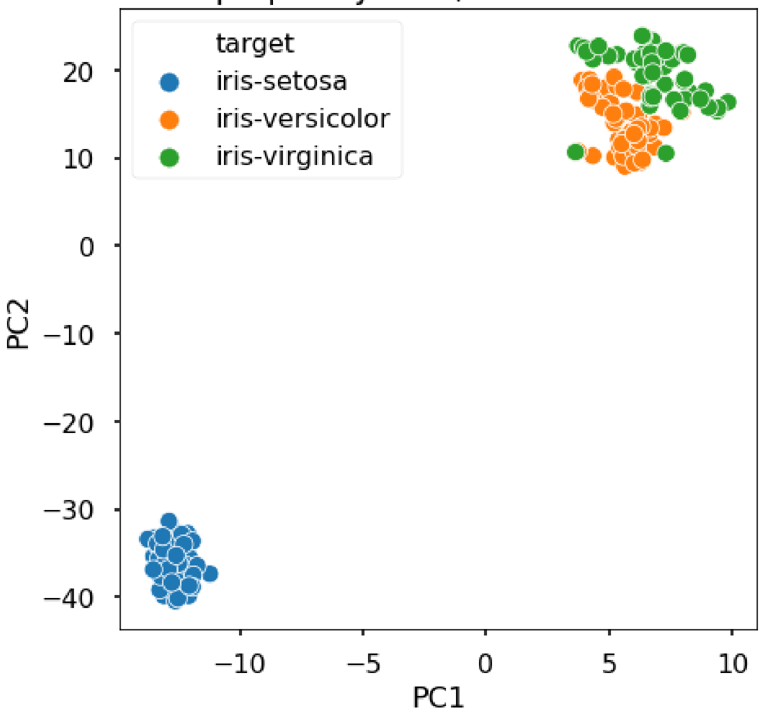

t-SNE visualization with perplexity -- 30, iterations -- 350 and epsilon -- 400
PC2
PC1
target
iris-setosa
iris-versicolor
iris-virginica
40
20
0
−20
−40
−30
−20
−10
0
10
20
30
40


t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 400
PC2
PC1
target
iris-setosa
iris-versicolor
iris-virginica
20
10
0
−10
−20
−30
−40
−15
−10
−5
0
5
10

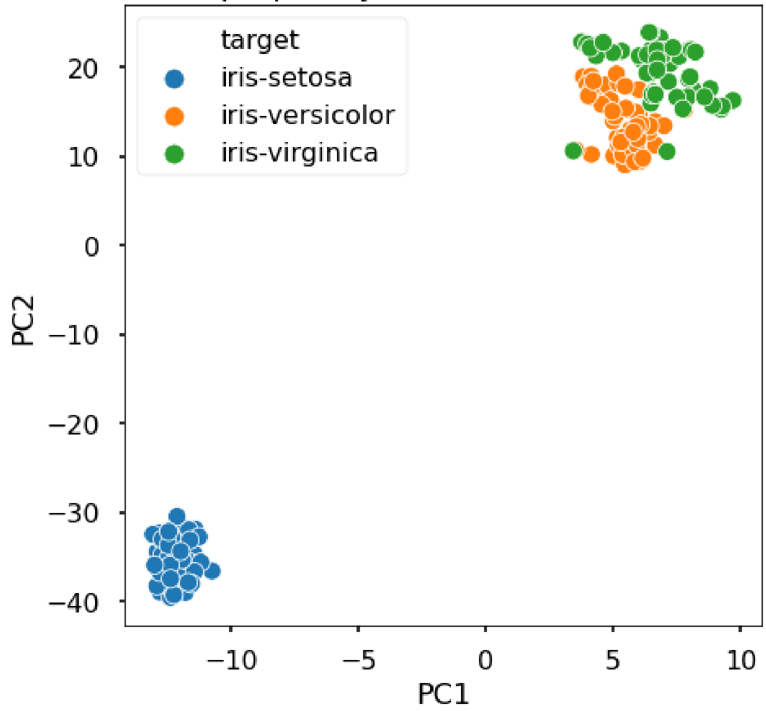
t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 400



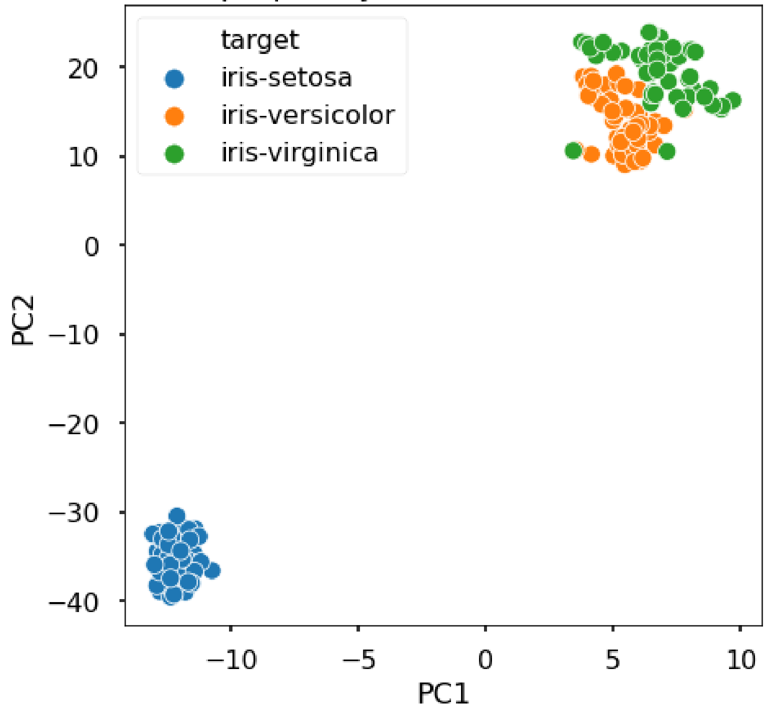
t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 400



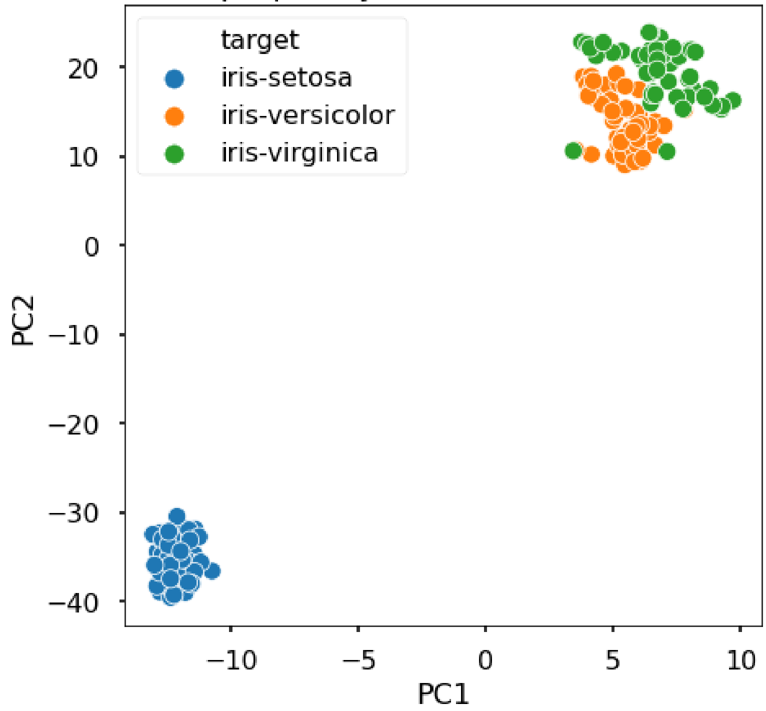
t-SNE visualization with perplexity -- 30, iterations -- 2500 and epsilon -- 400



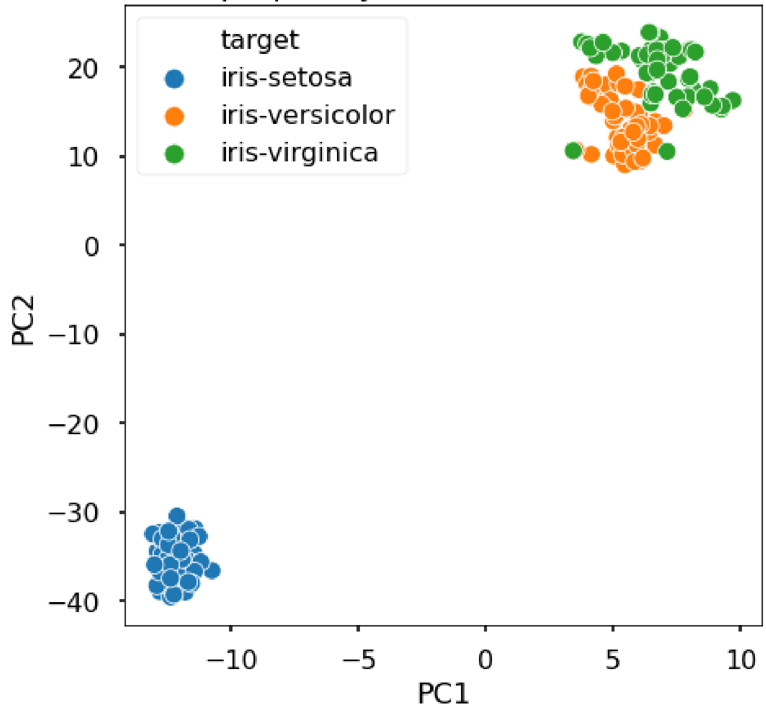
t-SNE visualization with perplexity -- 30, iterations -- 3500 and epsilon -- 400

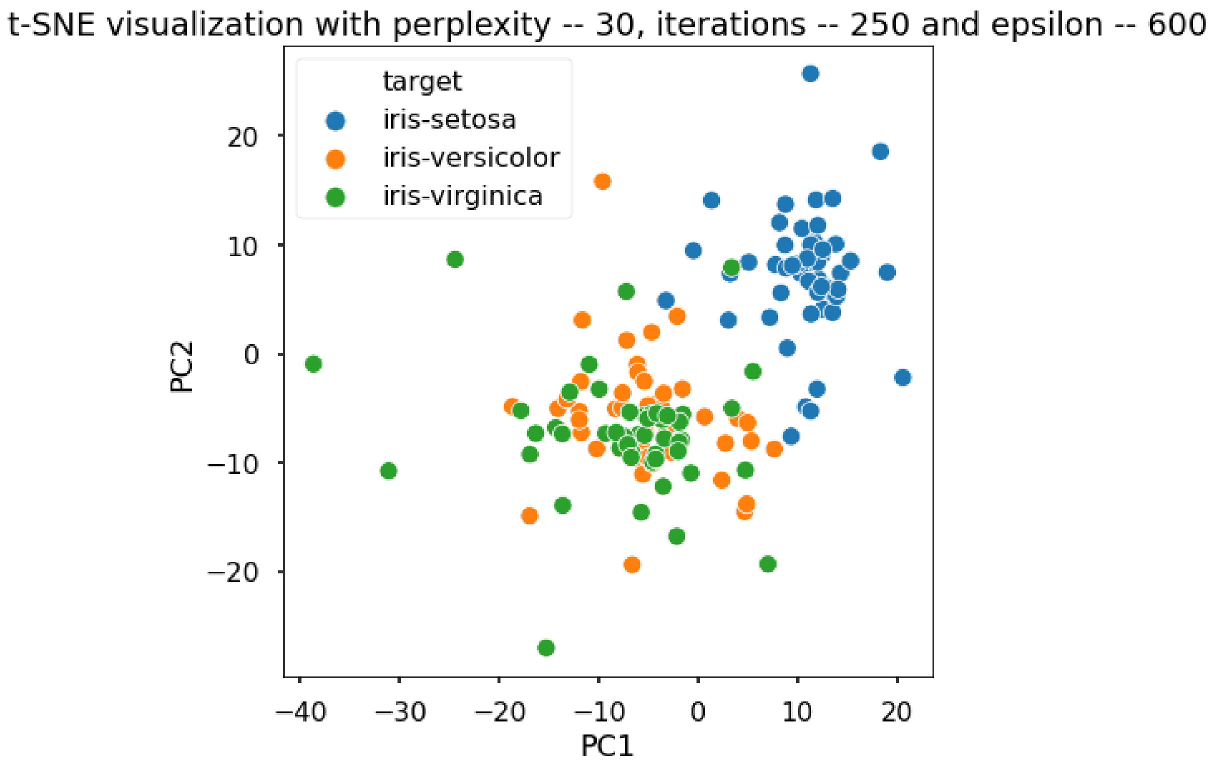


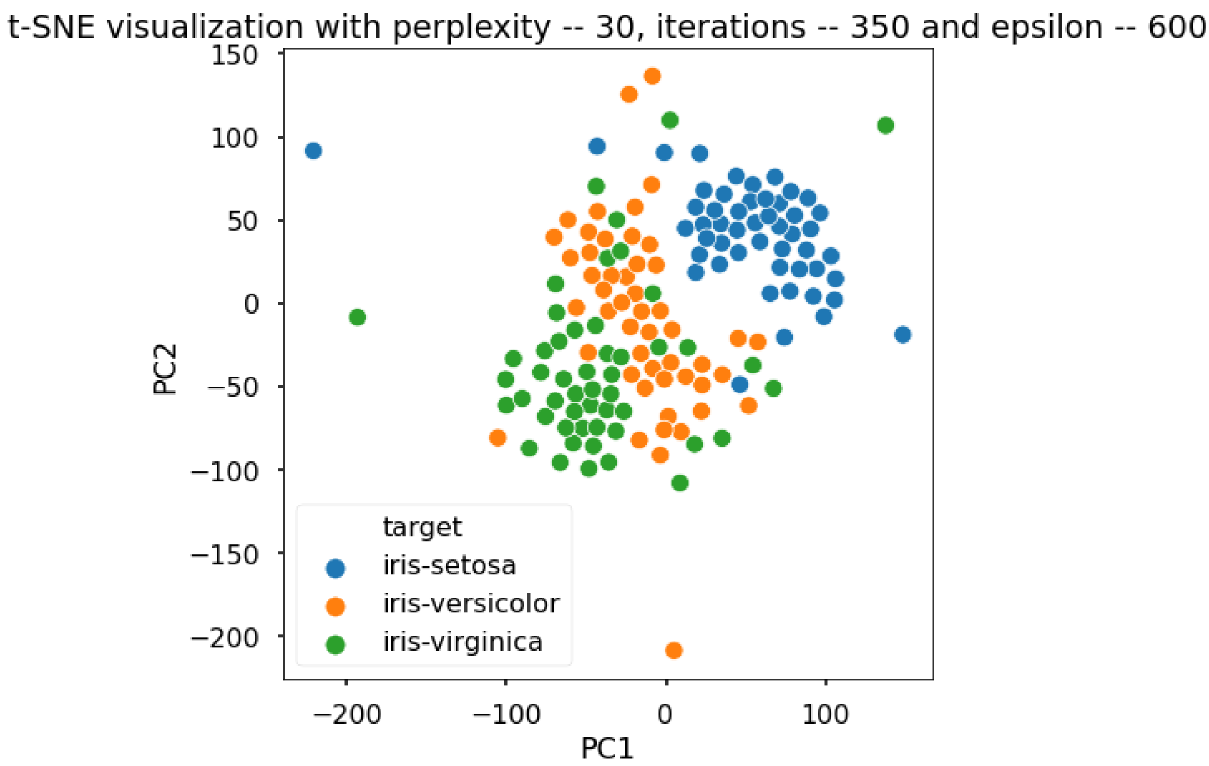
t-SNE visualization with perplexity -- 30, iterations -- 4000 and epsilon -- 400



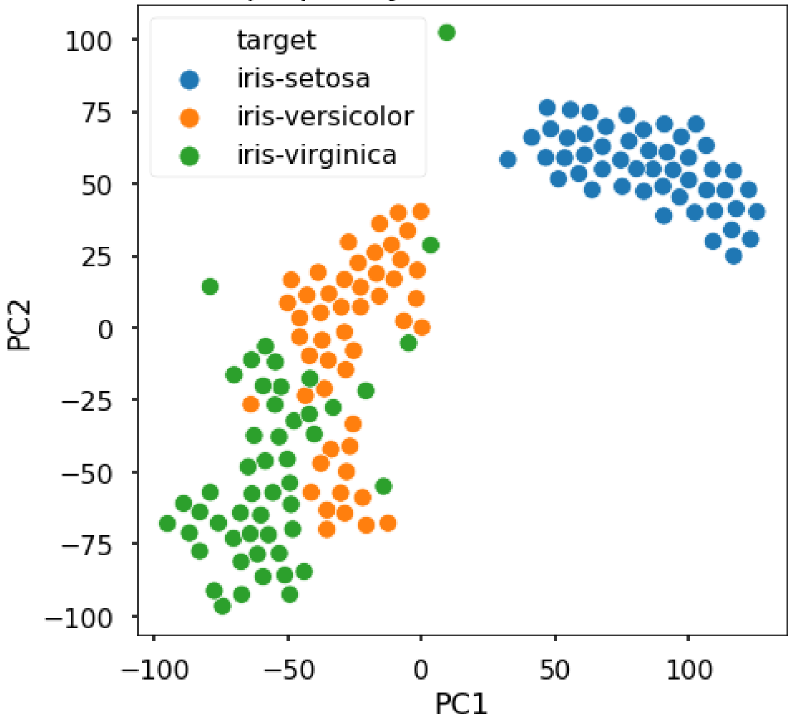
t-SNE visualization with perplexity -- 30, iterations -- 5000 and epsilon -- 400



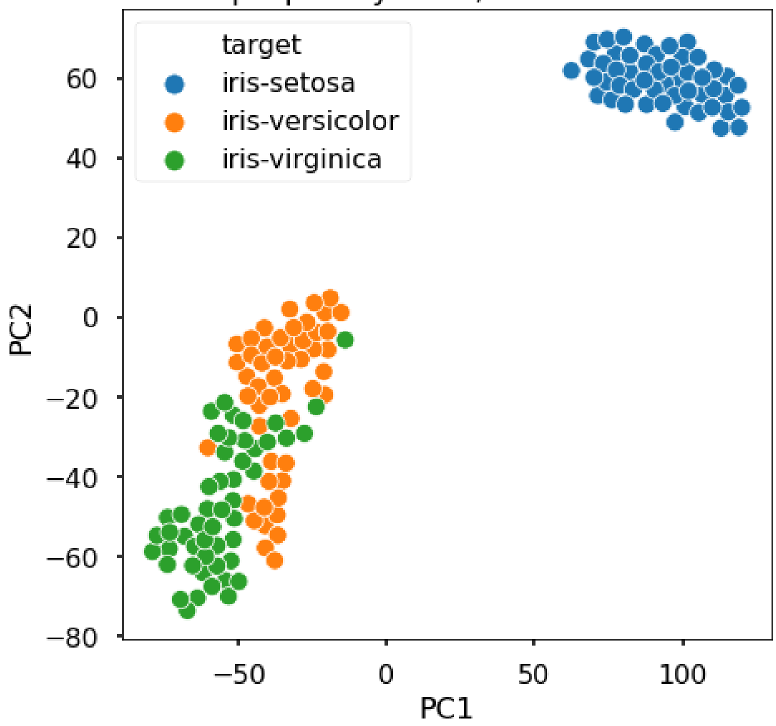

t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 600
target
iris-setosa
iris-versicolor
iris-virginica
PC2
20
10
0
−10
−20
−40
−30
−20
−10
0
10
20
PC1


t-SNE visualization with perplexity -- 30, iterations -- 350 and epsilon -- 600
150
100
50
0
−50
−100
−150
−200
PC2
target
iris-setosa
iris-versicolor
iris-virginica
−200
−100
0
100
PC1

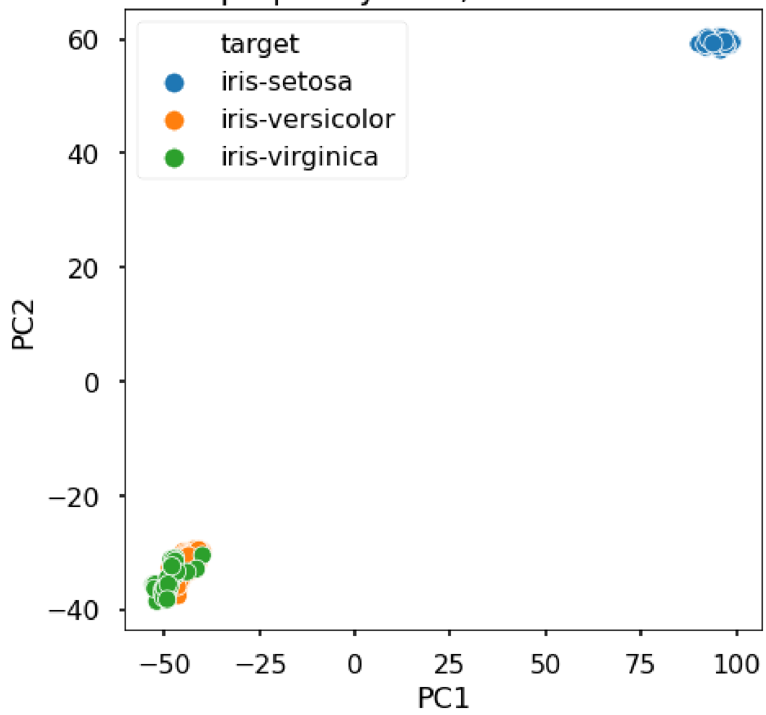
t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 600



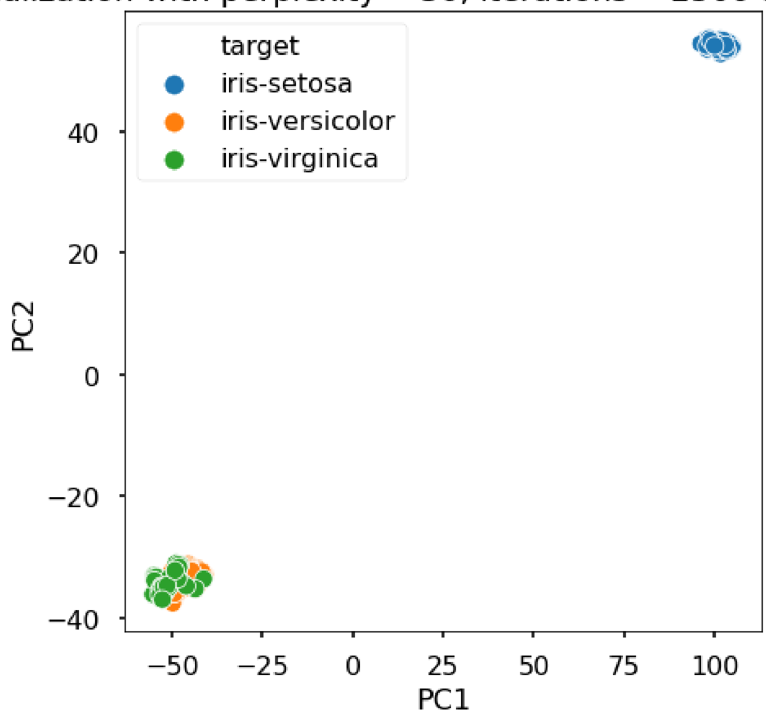
t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 600

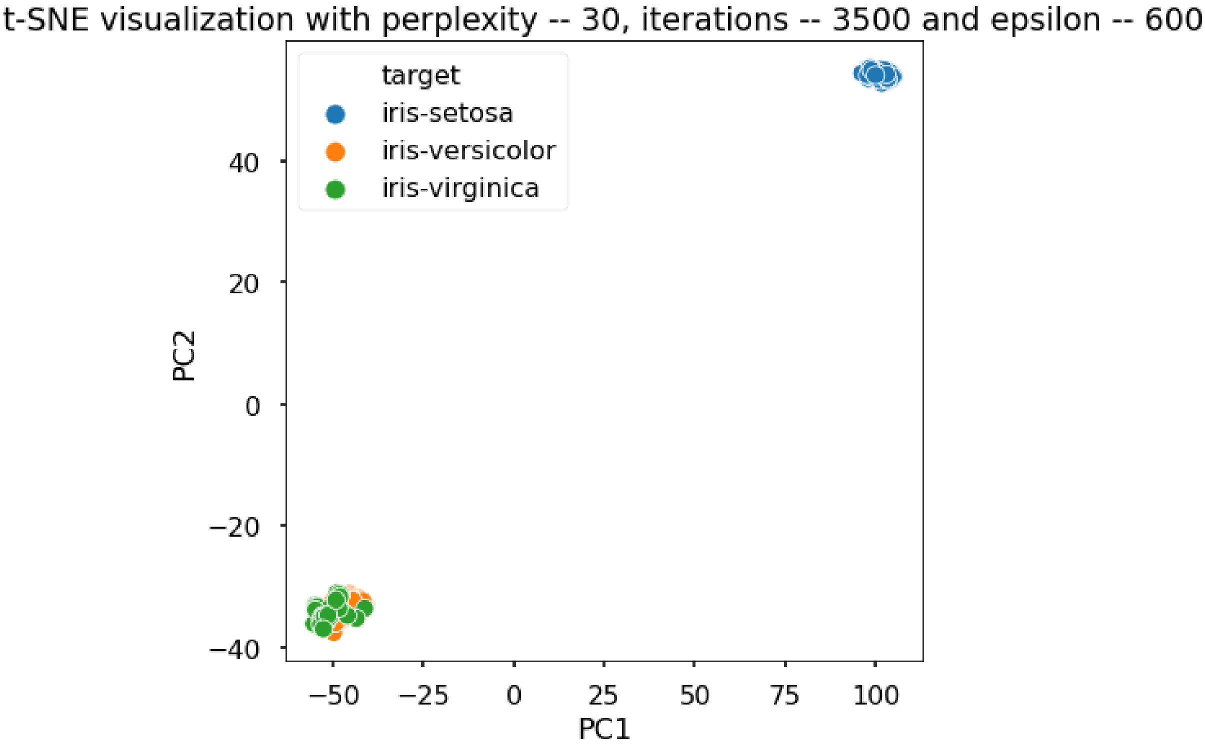


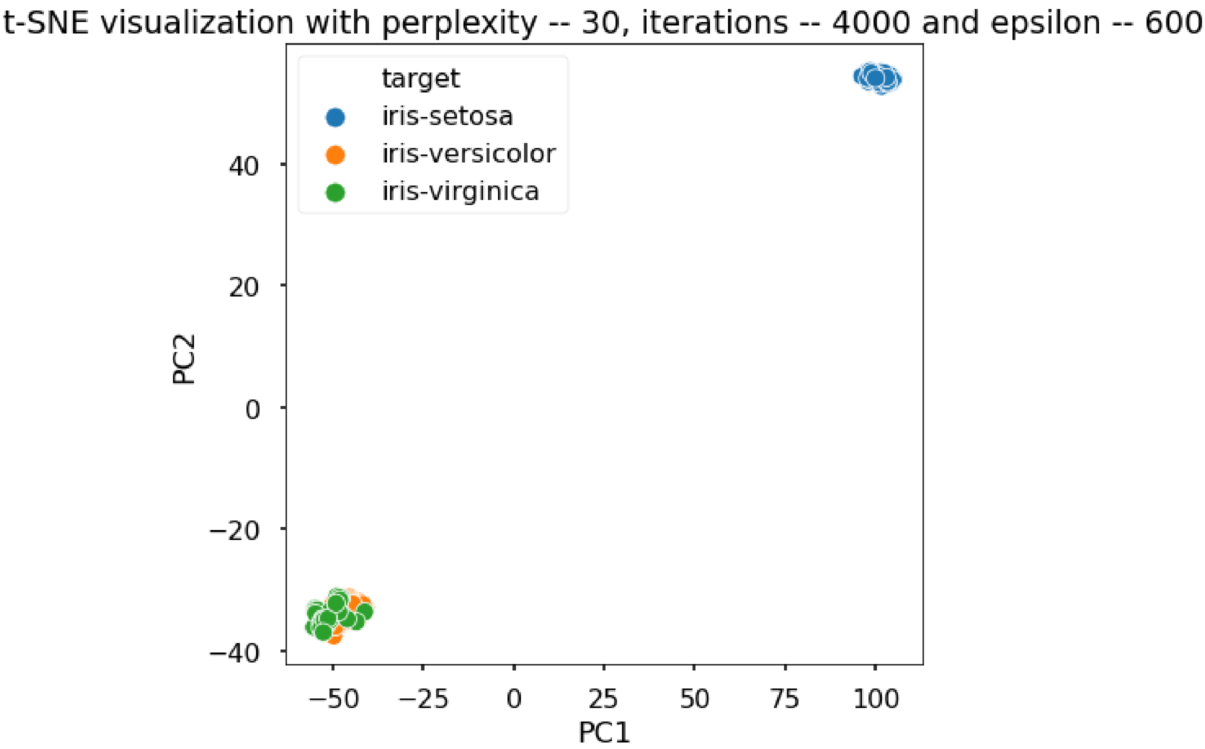
t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 600

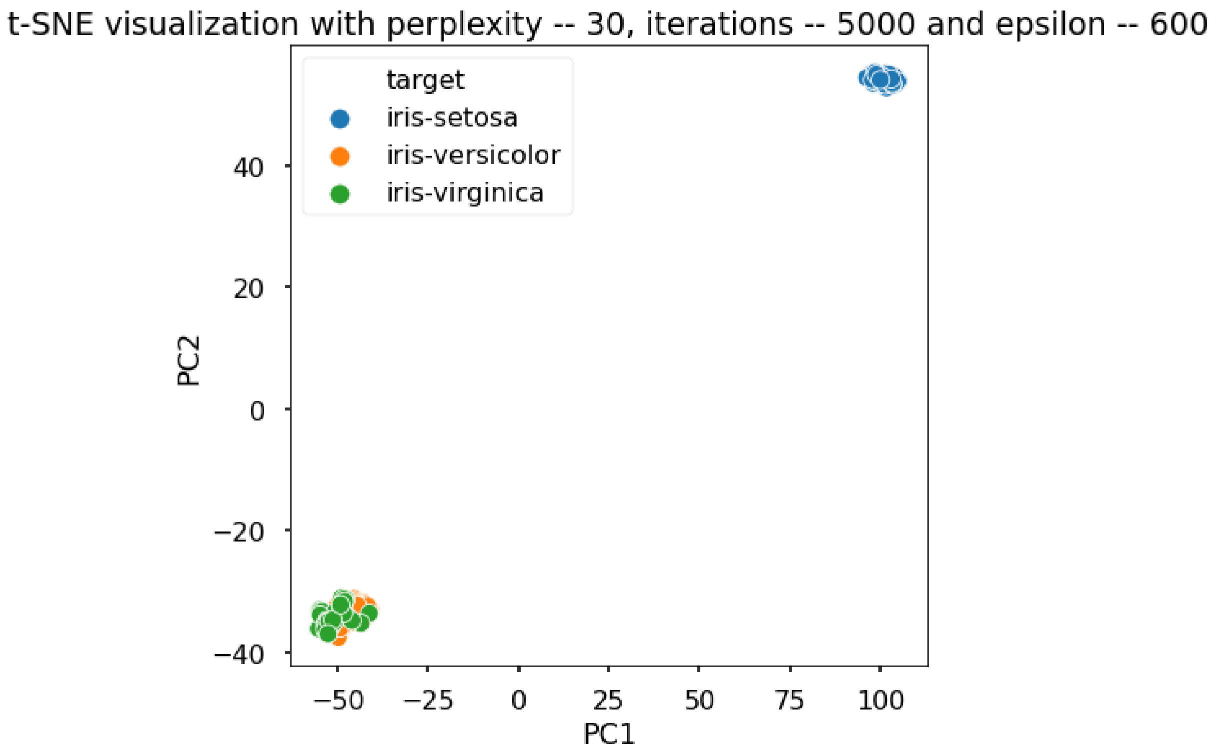


t-SNE visualization with perplexity -- 30, iterations -- 2500 and epsilon -- 600




t-SNE visualization with perplexity -- 30, iterations -- 3500 and epsilon -- 600
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
40
20
0
−20
−40
−50
−25
0
25
50
75
100


t-SNE visualization with perplexity -- 30, iterations -- 4000 and epsilon -- 600
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
40
20
0
−20
−40
−50
−25
0
25
50
75
100

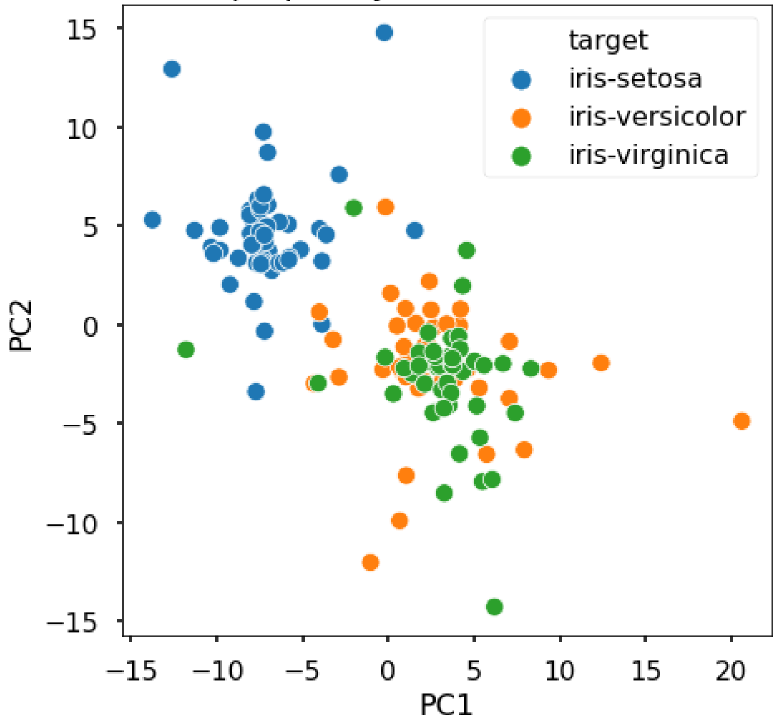


IRIS:CASE-III-Multiple_runs

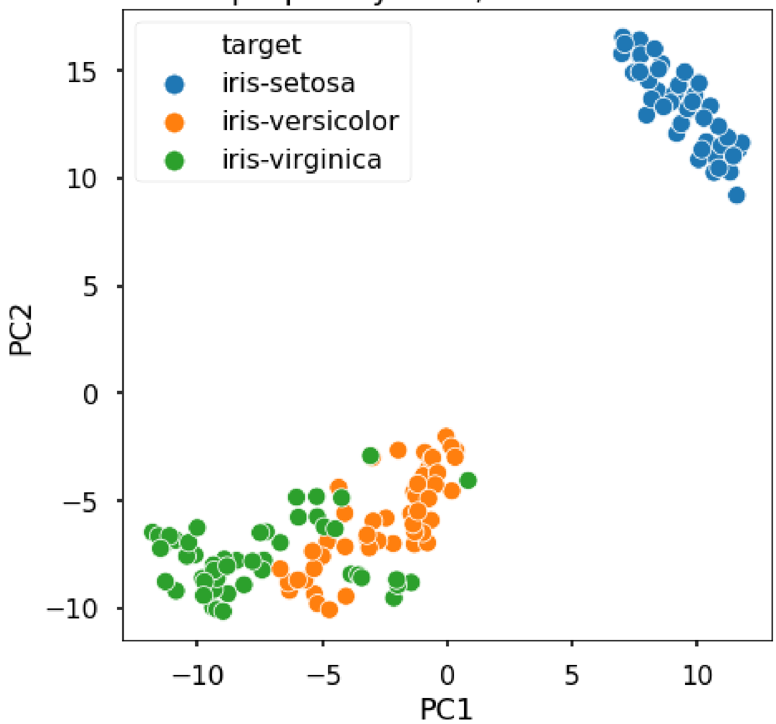
Running t-SNE on a range of iterations of fixed perplexity and learning rate(epsilon) with embedding initialization as 'random'

In [80]:
```
iterations = [250,350,500,750,1000,2500,3500,4000,5000]

for idx in range(len(iterations)):
    tsne_iris_1 = TSNE(n_components=2,perplexity=30,learning_rate=200,n_iter=iterati
                       init='random',n_jobs=-1)
    iris1_tsne_pcmps = pd.DataFrame(tsne_iris_1.fit_transform(iris_stand_df.iloc[:,0
    iris1_tsne_pcmps = pd.concat([iris1_tsne_pcmps,iris_stand_df['target']],axis=1)
    with plt.style.context('seaborn-poster'):
        plt.figure(figsize=(7,7))
        sns.scatterplot(data=iris1_tsne_pcmps,x='PC1',y='PC2',hue='target')
        plt.title("t-SNE visualization with perplexity -- {0}, iterations -- {1} and


        plt.show()
```

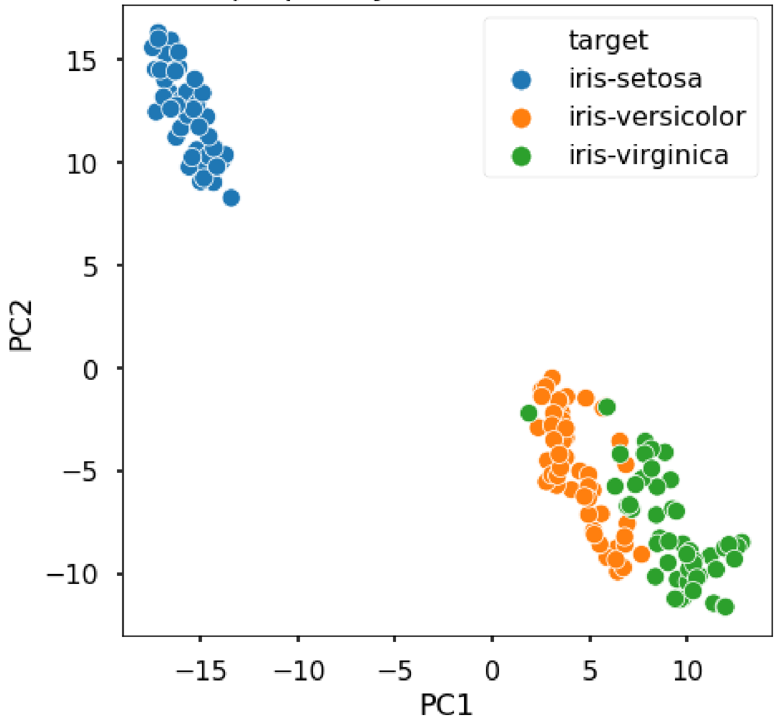
t-SNE visualization with perplexity -- 30, iterations -- 250 and epsilon -- 200



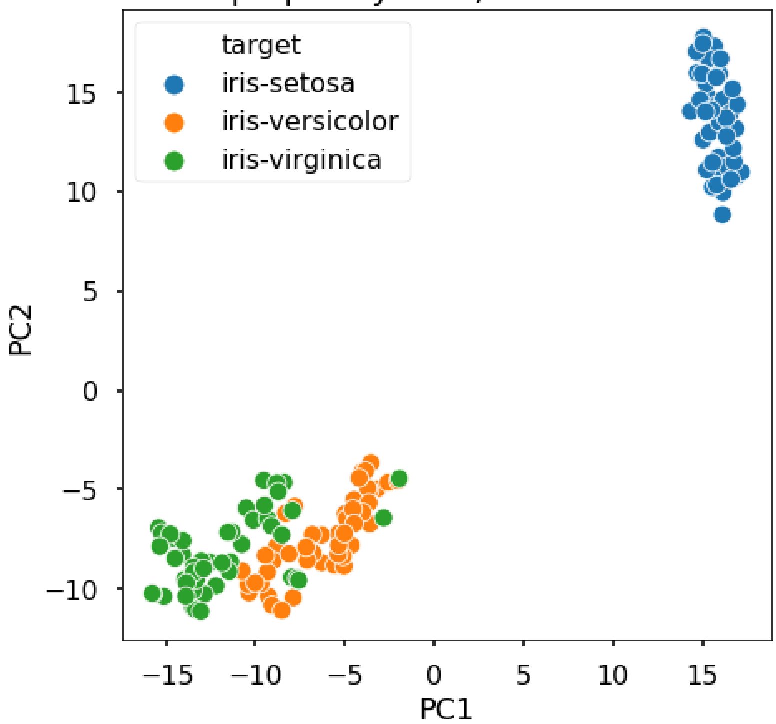
t-SNE visualization with perplexity -- 30, iterations -- 350 and epsilon -- 200

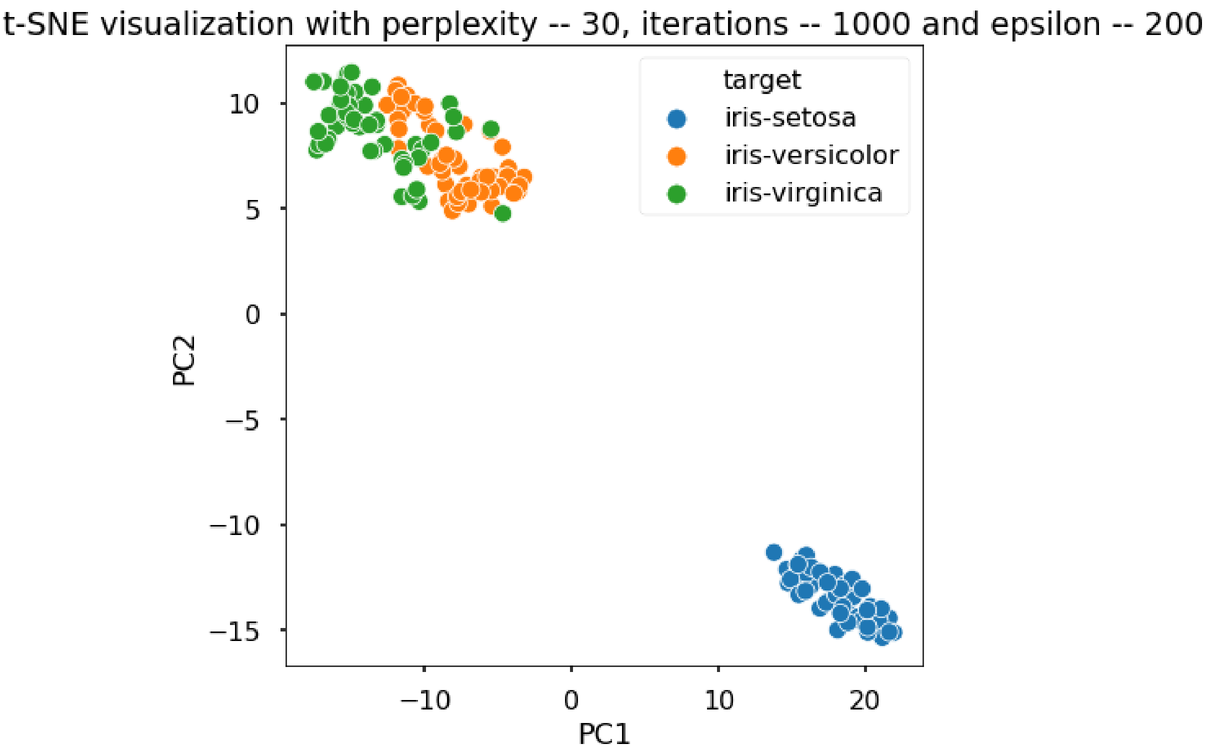


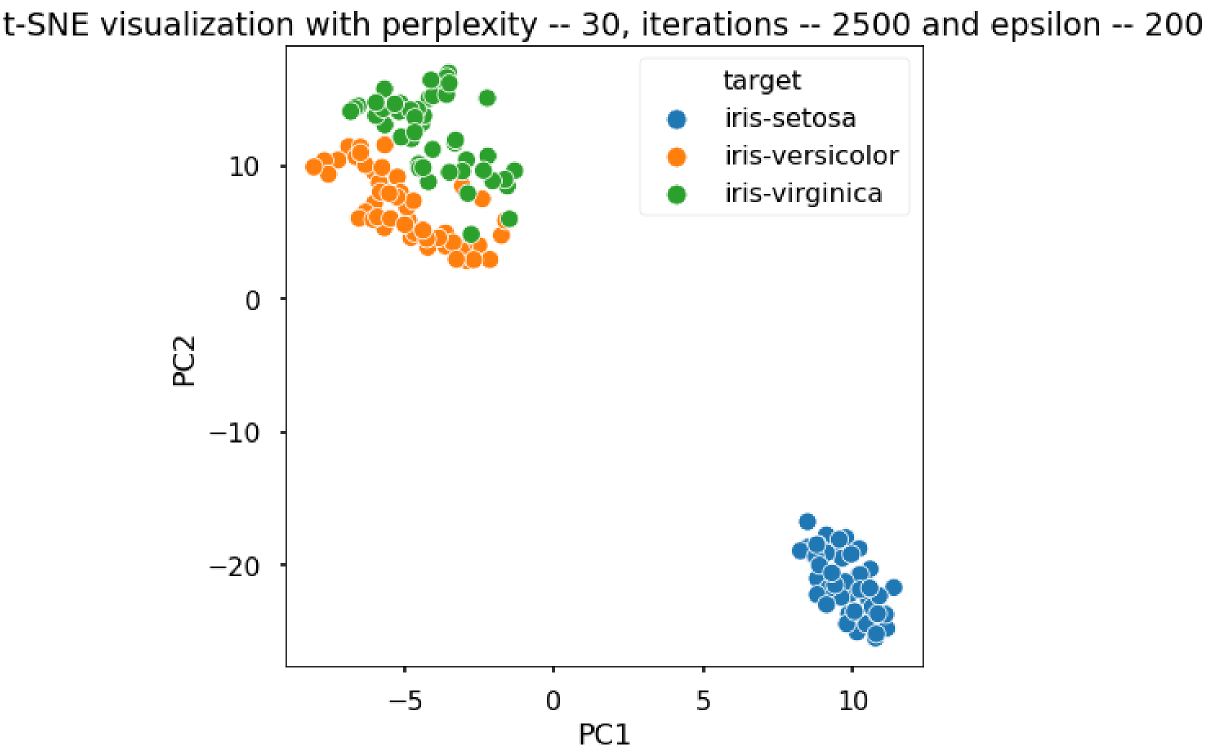
t-SNE visualization with perplexity -- 30, iterations -- 500 and epsilon -- 200

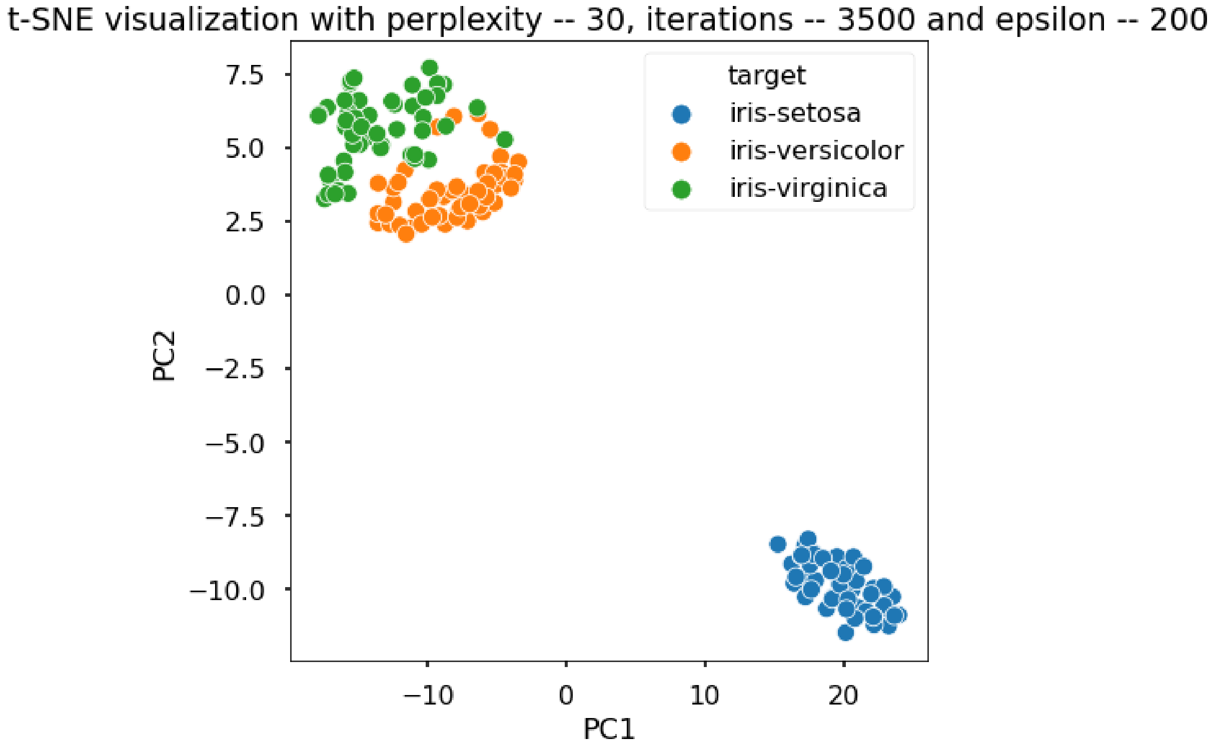


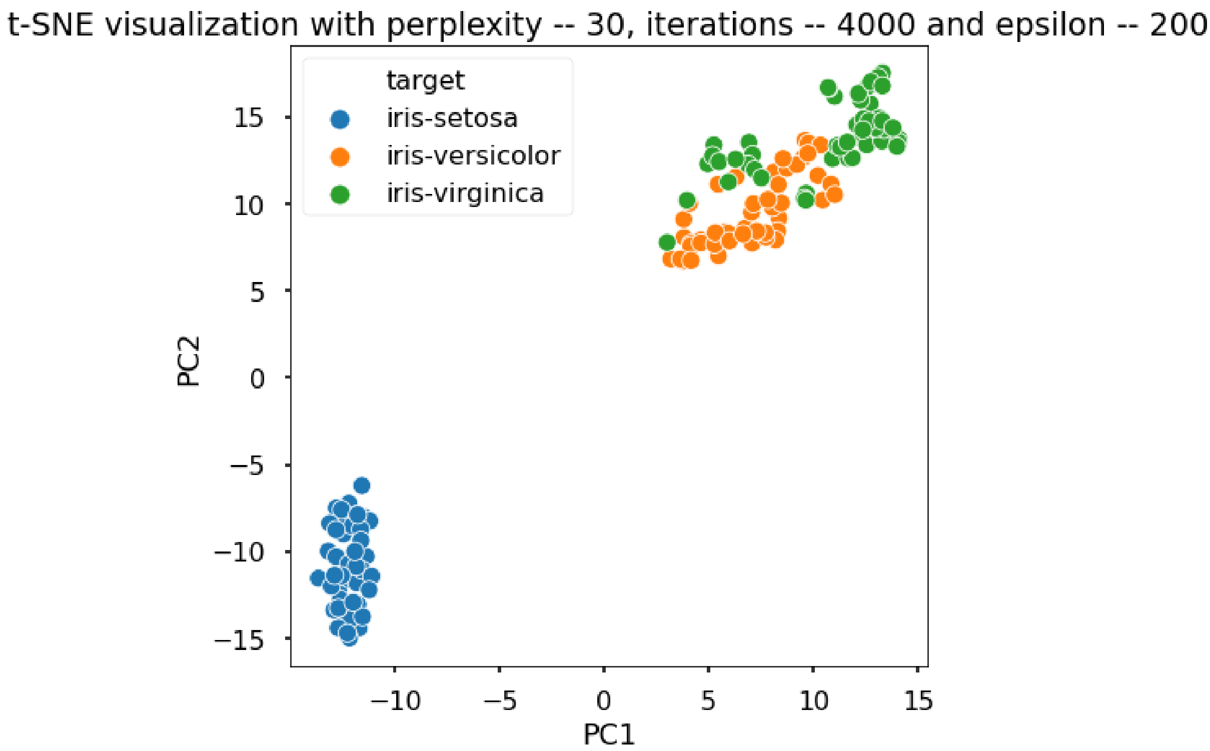
t-SNE visualization with perplexity -- 30, iterations -- 750 and epsilon -- 200

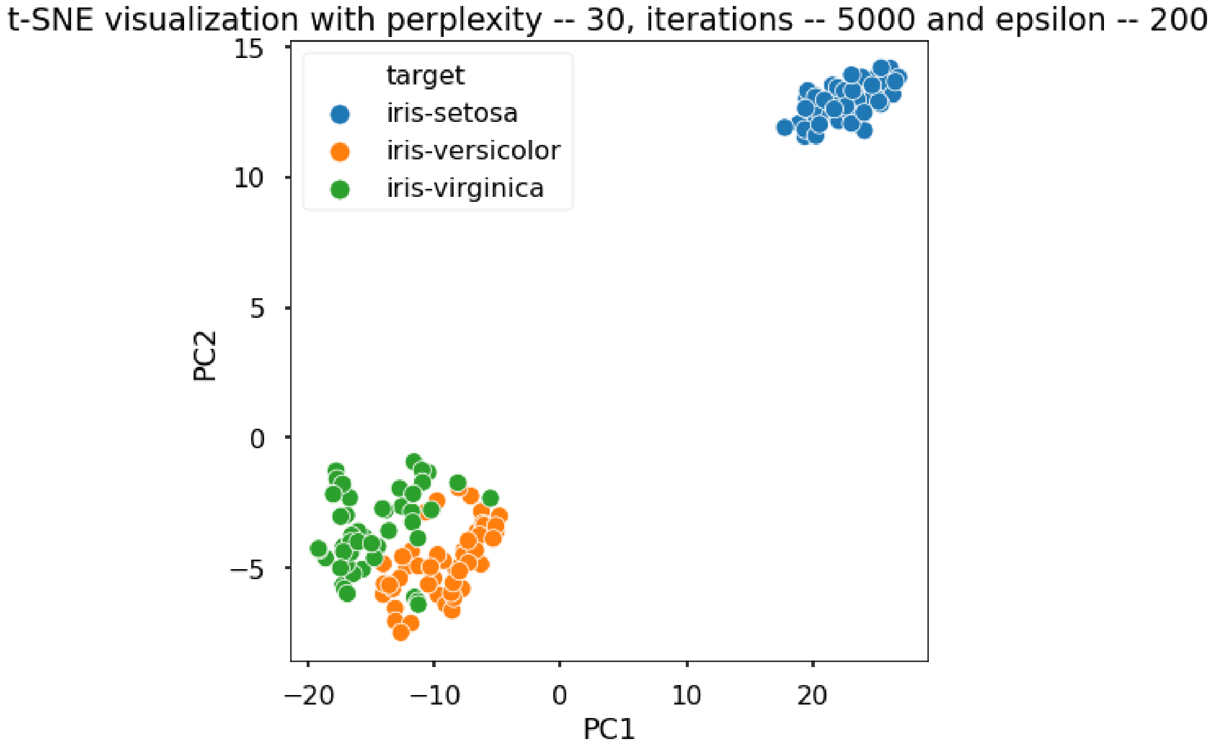



t-SNE visualization with perplexity -- 30, iterations -- 1000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
10
5
0
−5
−10
−15
−10
0
10
20


t-SNE visualization with perplexity -- 30, iterations -- 2500 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
PC2
PC1
10
0
−10
−20
−5
0
5
10


t-SNE visualization with perplexity -- 30, iterations -- 3500 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
PC2
−10
0
10
20
PC1


t-SNE visualization with perplexity -- 30, iterations -- 4000 and epsilon -- 200
target
iris-setosa
iris-versicolor
iris-virginica
15
10
5
0
−5
−10
−15
PC2
−10
−5
0
5
10
15
PC1



IRIS:Result_of_multiple_runs

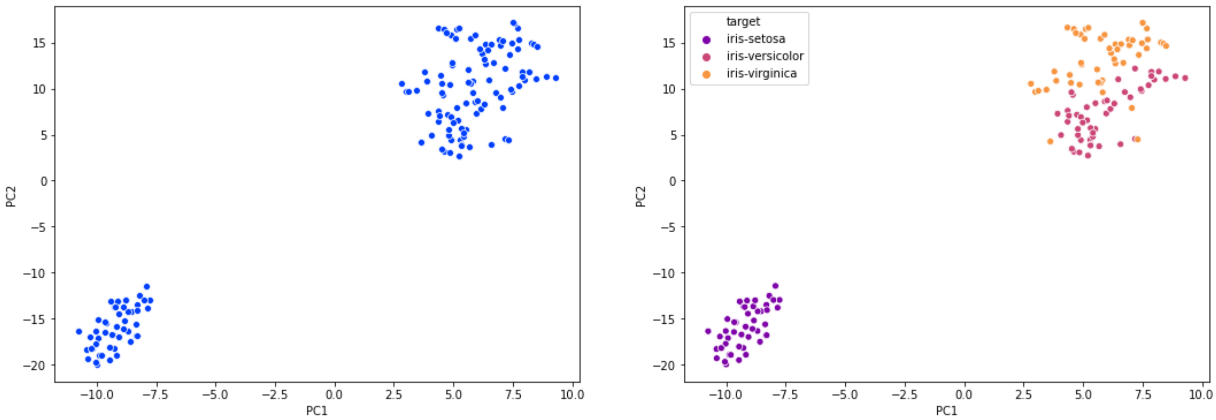
In [86]:
```
tsne_iris_1 = TSNE(n_components=2,perplexity=30,learning_rate=200,n_iter=5000,init='
```

In [87]:
```
iris_tsne_cmps = pd.concat([pd.DataFrame(tsne_iris_1.fit_transform(iris_stand_df.ilo
                            pd.DataFrame(iris_stand_df.target,columns=['target'])],a
iris_tsne_cmps.head()
```

Out[87]:

	PC1	PC2	target
0	-9.326900	-16.767998	iris-setosa
1	-9.125493	-13.043600	iris-setosa
2	-8.535302	-14.208212	iris-setosa
3	-8.310025	-13.506122	iris-setosa
4	-9.926417	-17.123808	iris-setosa

In [89]:
```
with plt.style.context('seaborn-bright'):
    fig,ax = plt.subplots(nrows=1,ncols=2,figsize=(18,6))
    sns.scatterplot(data=iris_tsne_cmps,x='PC1',y='PC2',ax=ax[0])
    sns.scatterplot(data=iris_tsne_cmps,x='PC1',y='PC2',hue='target',ax=ax[1],palett
```



Other_method

IRIS : t-SNE with 'pca' initialization embedding, Nearest neighbors error method as 'exact' and distance metric as 'minkowski'

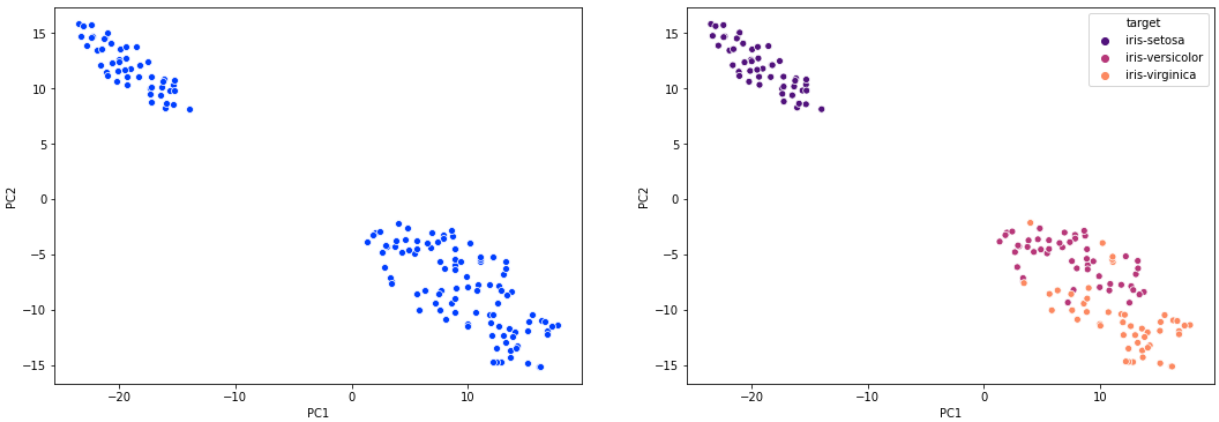
In [92]:
```
import scipy

tsne_iris_rand = TSNE(n_components=2,
                      perplexity=30,
                      learning_rate=200,
                      n_iter=5000,
                      init='pca',                ## Good with the global capturing
                      method='exact',            ## This method is slower and not s
                      metric=scipy.spatial.minkowski_distance,
                      n_jobs=-1)
```

In [93]:
```
iris_tsne_cmps = pd.concat([pd.DataFrame(tsne_iris_rand.fit_transform(iris_stand_df.
                           pd.DataFrame(iris_stand_df.target,columns=['target'])],a
iris_tsne_cmps.head()
```

Out[93]:

	PC1	PC2	target
0	-20.068819	12.580353	iris-setosa
1	-15.324358	10.337151	iris-setosa
2	-17.385283	9.968454	iris-setosa
3	-16.503378	9.369566	iris-setosa
4	-19.959255	13.517425	iris-setosa

In [94]:
```
with plt.style.context('seaborn-bright'):
    fig,ax = plt.subplots(nrows=1,ncols=2,figsize=(18,6))
    sns.scatterplot(data=iris_tsne_cmps,x='PC1',y='PC2',ax=ax[0])
    sns.scatterplot(data=iris_tsne_cmps,x='PC1',y='PC2',hue='target',ax=ax[1],palett
```



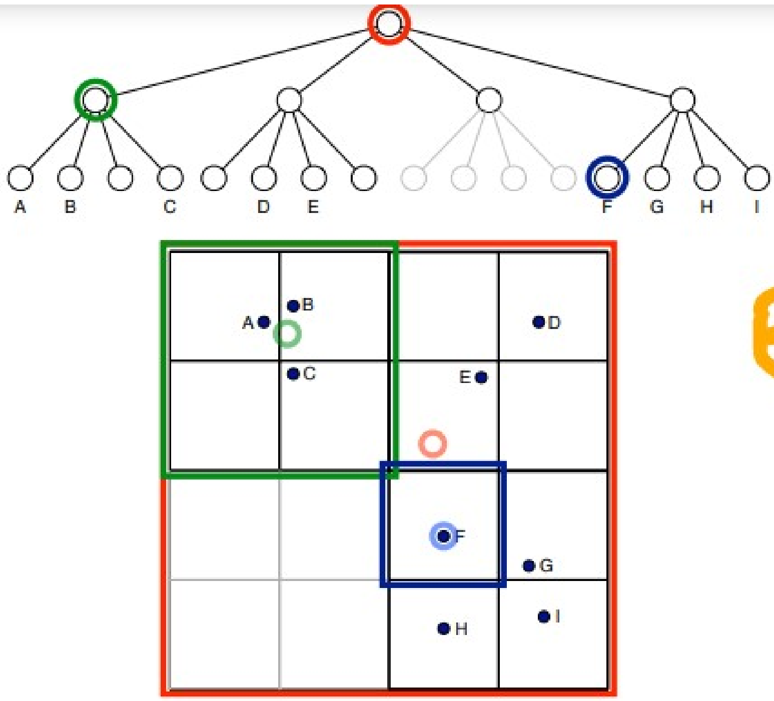
BH_DT_Approx

Some detailing of how Barnes-Hut & Dual Tree Faster approximations of t-SNE works?

In [1]:
```
from IPython.display import Image
```

In [2]:
```
# How Barnes-Hut Approx performs point-cell interaction using single depth-first sea
Image("Refer_Notes/TSNE_Barnes_Hut_Approx.jpg",width=700,height=700)
```

Out[2]:



B·H

Figure 1: Illustration of a quadtree that was constructed on a data set of nine two-dimensional data points. The top half of the figure illustrates the structure of the tree that represents the partitioning of the two-dimensional space shown in the lower half of the figure. Corresponding colors are used to highlight corresponding elements of the graph and the space partitioning. Nodes in the graph correspond to square cells in the space (deeper nodes correspond to smaller cells). In each node, we store: (1) the number of points that are located in the corresponding cell and (2) the center-of-mass of those points (the centers-of-mass of the three highlighted cells are illustrated by the opaque circles in the space partitioning). The opaque parts of the tree are not actually created, because the corresponding parts of the space do not contain any data points. Leaf nodes represent cells that contain at most one data point. As a result, denser areas of the space correspond to parts of the tree that are deeper.

In [3]:
```
# How Barnes-Hut Approx Summary Condition
Image("Refer_Notes/TSNE_Barnes_Hut_Approx_Summary_Condition.jpg",width=700,height=70
```

Out[3]:

We use the condition proposed by Barnes and Hut (1986) to decide whether a cell may be used as a "summary" for all points in that cell. The condition compares the distance between the cell and the target point with the size of that cell:

$$\frac{r_{cell}}{||\mathbf{y}_i - \mathbf{y}_{cell}||^2} < \theta, \tag{9}$$

where r_{cell} represents the length of the diagonal of the cell under consideration and θ is a threshold that trades off speed and accuracy (higher values of θ lead to faster but coarser approximations). Note that when $\theta = 0$, all pairwise interactions are computed, and the Barnes-Hut approximation reduces to naive computation of the t-SNE gradient. In preliminary experiments, we also explored various other conditions that take into account the rapid decay of the Student-t tail, but we did not find these alternative conditions to lead to a better accuracy-speed trade-off. The problem of more complex conditions is that they require expensive computations at each cell. By contrast, the condition in Equation 9 can be evaluated very rapidly.

In [4]:
```
# How Dual-Tree Approx performs cell-cell interaction using two depth-first search o
Image("Refer_Notes/TSNE_Dual_Tree_Approx.jpg",width=700,height=700)
```

Out[4]:

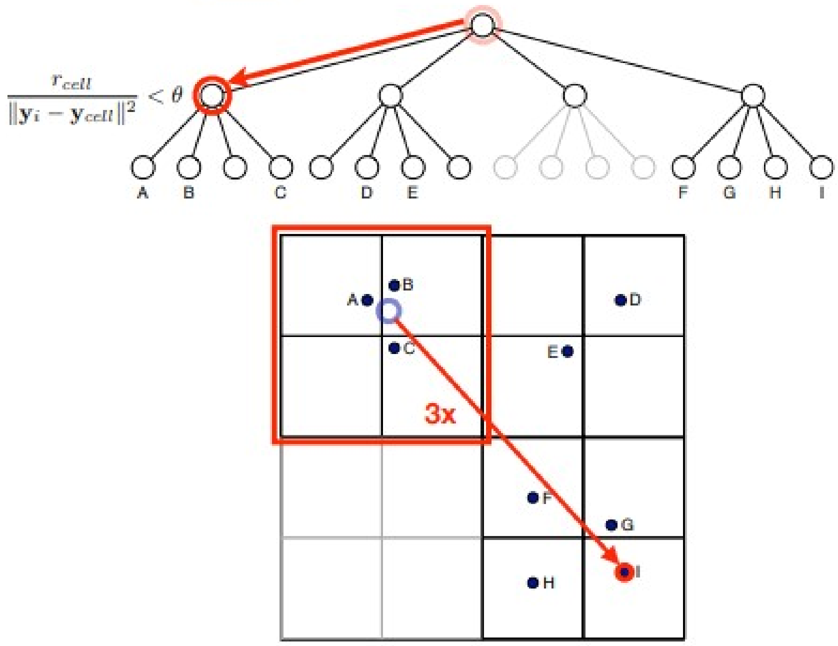


Figure 2: Illustration of the Barnes-Hut approximation. To evaluate the t-SNE gradient for point I, the Barnes-Hut algorithm performs a depth-first search on the embedding quadtree, checking at every node whether or not the node may be used as a "summary". In the illustration, the cell containing points A, B, and C satisfies the summary-condition: the force between the center-of-mass of the three points (which is stored in the quadtree node) and point I is computed, multiplied by the number of points in the cell (*i.e.*, by three), and added to the gradient for point I. All children of the summary node are pruned from the depth-first search.

cells. As before, we compute the attractive part of the t-SNE gradient in Eqn. 8 exactly in dual-tree t-SNE. However, in dual-tree t-SNE, the dual-tree algorithm is used to compute the repulsive part, F_{rep}, of the t-SNE gradient. Note that the optimal value for θ generally differs between Barnes-Hut and dual-tree algorithms, because both algorithms summarize interactions differently.

In [5]:
```
# How Dual-Tree Approx summary consition
Image("Refer_Notes/TSNE_Dual_Tree_Approx_Summary_Condition.jpg",width=700,height=700
```

Out[5]:

4.3 Dual-tree Approximation

Whilst the Barnes-Hut algorithm considers *point-cell* interactions, further speed-ups may be obtained by computing only *cell-cell* interactions. This can be done using a dual-tree algorithm of Gray and Moore (2001). The dual-tree algorithm simultaneously traverses the same quadtree twice in a depth-first manner. For every pair of nodes, the dual-tree algorithm decides whether or not the interaction between the cells of quadtree A and quadtree B can be used as "summary" for the interactions between all points inside these two cells (note that quadtree A and B are identical trees). If the summary condition is passed, the corresponding force is computed. Subsequently, we perform the following additions: (1) we add to all children of the node under consideration in tree A the product of the force and the number of children in the relevant node of tree B; and (2) we add to all children of the node under consideration in tree B the product of the force and the number of children in the node of tree A. Subsequently, all children of the cells in quadtree A and B are pruned. In the dual-tree approximation, we use the following condition to check whether the interaction between a pair of nodes may be used as a "summary" interaction:

$$\frac{\max(r_{cell-A}, r_{cell-B})}{\|\mathbf{y}_{cell-A} - \mathbf{y}_{cell-B}\|^2} < \theta, \tag{10}$$

where $\mathbf{y}_{cell-A}$ and $\mathbf{y}_{cell-B}$ represent the center-of-mass of the two cells from quadtree A and B under consideration and where r_{cell-A} and r_{cell-B} represent the diameter of these two

In [6]:
```python
# How Barnes-Hut and Dual-Tree comparison
Image("Refer_Notes/TSNE_Dual_Tree_Barnes_Hut_Comparison.jpg",width=700,height=700)
```

Out[6]:

Whilst the dual-tree algorithm may lead to significant reductions in the number of pairwise forces that needs to be computed compared to the Barnes-Hut algorithm, the computational advantages of the dual-tree algorithm are smaller than one might initially expect when the dual-tree algorithm is used to approximate the t-SNE gradient. Specifically, the problem is that after computing an interaction between two cells, one still needs to determine to which set of points the interaction applies. That is, we need to perform an additional search to determine which points are located in the cell corresponding to the

10

Accelerating t-SNE using Tree-Based Algorithms

nodes under consideration (in both tree A and B), because the force needs to be added to all those points (after multiplication with the appropriate number of children). Alternatively, we could construct and store a list of all children for each node during tree construction, but this is computationally equally costly and requires substantial additional memory[4].

Conclusion

In my analysis with two different dataset I found that t-SNE gives better results when embedding

method is 'PCA' inplace of 'random' and nearest neighbors error method as 'exact' instead of 'barnes_hut'.

The 'barnes_hut' method is much better in terms of execution performance as compared to 'exact' and it is not a good approach with a large dataset. If you want the nearest neighbors error to be less than 3% and your dataset is not very large then better go with 'exact'.