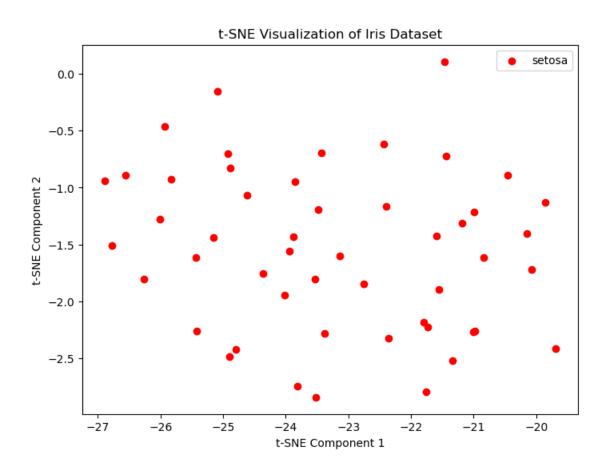
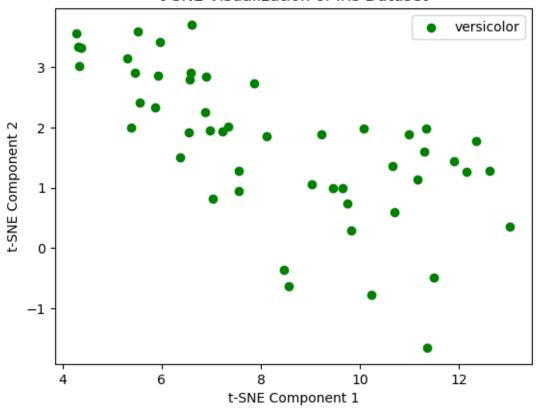
## T-SNE algorithm

## December 3, 2023

```
[8]: | #Importing required libraries for the t-SNE visualization
      #scikit-learn's datasets module for loading the iris dataset
      #t-SNE (t-distributed Stochastic Neighbor Embedding) from scikit-learn for
      ⇔dimensionality reduction
      #Matplotlib for plotting
      from sklearn import datasets
      from sklearn.manifold import TSNE
      import matplotlib.pyplot as plt
 [9]: #Loading the Iris dataset using scikit-learn's datasets module
      iris = datasets.load_iris()
[10]: #Extracting features (X) and target variable (y) from the Iris dataset
      X = iris.data
      y = iris.target
[11]: #Creating an instance of the t-SNE class with 2 components and a fixed random
      tsne = TSNE(n_components=2, random_state=42)
      X_tsne = tsne.fit_transform(X)
[12]: #Creating a scatter plot for the t-SNE visualization
      #Each point is colored based on the target variable (species of Iris)
      plt.figure(figsize=(8, 6))
      for i, c in zip(range(3), ['r', 'g', 'b']):
          plt.scatter(X_tsne[y == i, 0], X_tsne[y == i, 1], c=c, label=iris.
       →target_names[i])
          plt.xlabel('t-SNE Component 1')
          plt.ylabel('t-SNE Component 2')
          plt.title('t-SNE Visualization of Iris Dataset')
          plt.legend()
          plt.show()
```



## t-SNE Visualization of Iris Dataset



## t-SNE Visualization of Iris Dataset

