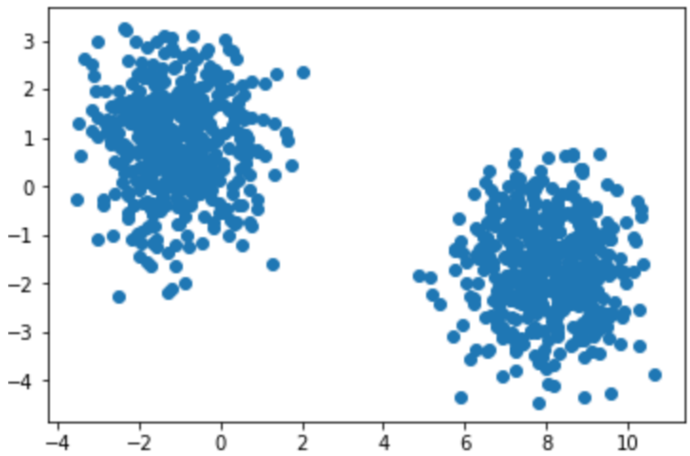
Lesson 1: Introduction to Clustering

Figure 1.1: Differences between unsupervised and supervised learning



Figures 1.2: Two distinct scatterplots

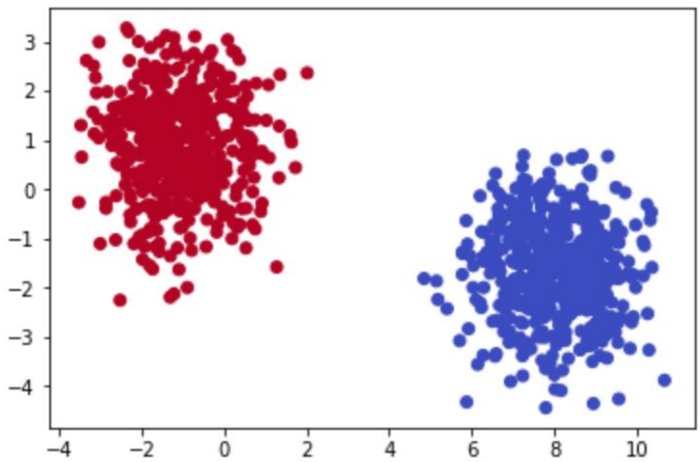


Figure 1.3: Scatterplots clearly showing clusters that exist in a provided dataset

```
array([[-0.72690901,  2.76012303],
       [-1.38504876,  2.16558784],
       [-1.12519969,  0.78279526],
       ...,
       [-0.92272983, -0.44782031],
       [ 8.26124228, -0.37099837],
       [-1.01204517,  0.3228703 ]])
```

Figures 1.4: Two-dimensional raw data in a NumPy array

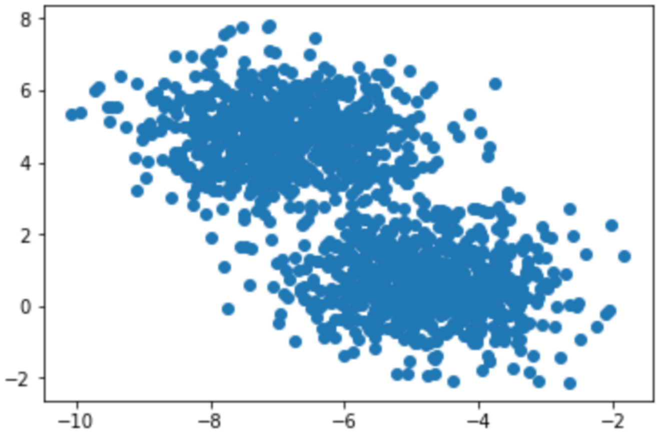


Figure1.5 Two-dimensional scatterplot

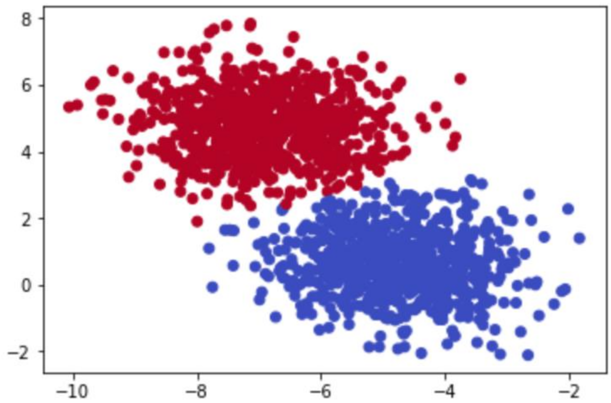


Figure 1.6: Clusters in the scatterplot

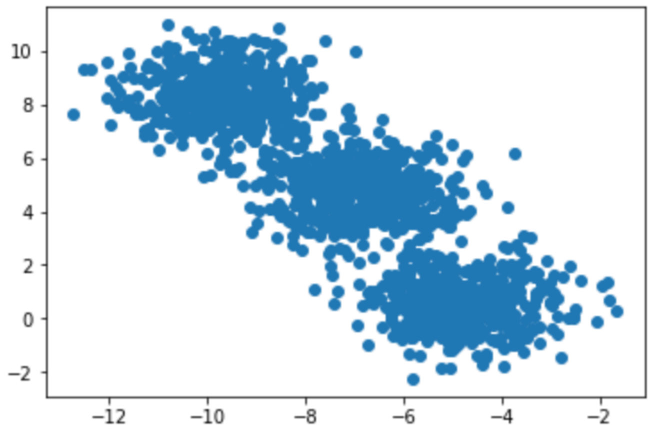


Figure1.7: Two-dimensional scatterplot

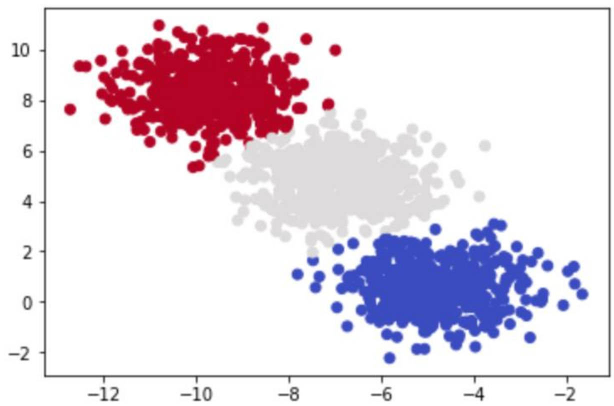


Figure 1.8: Clusters in the scatterplot

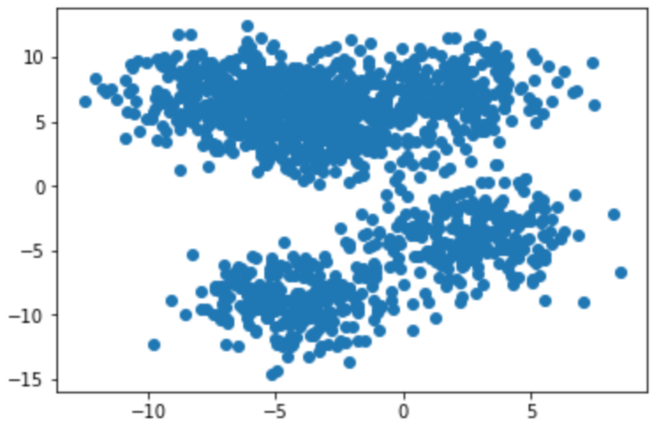


Figure1.9: Two-dimensional scatterplot

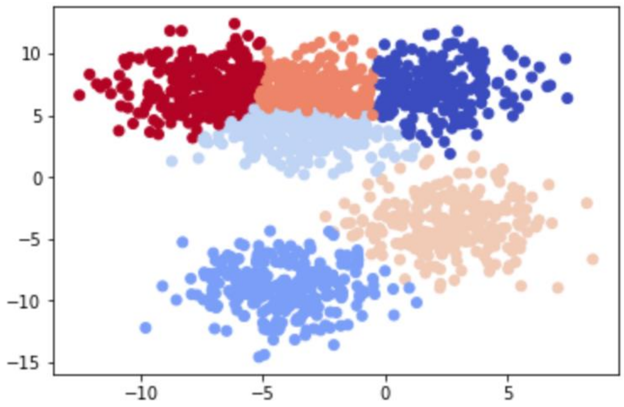


Figure 1.10: Clusters in the scatterplot

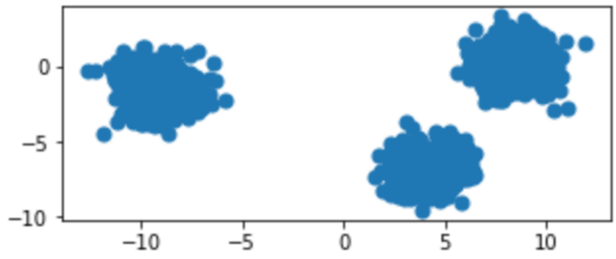


Figure 1.11: Original raw data charted on x,y coordinates

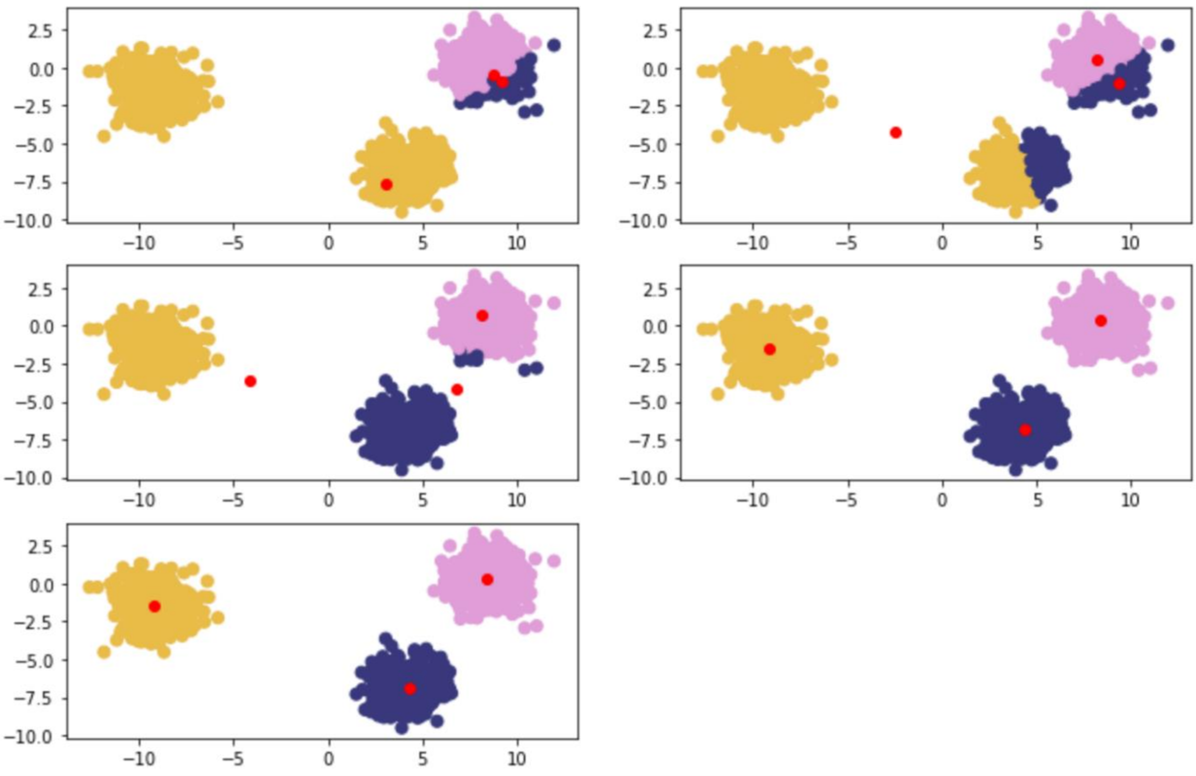


Figure 1.12: Reading from left to right – red points are randomly initialized centroids, and the closest data points are assigned to groupings of each centroid

$$d((x,y),(a,b)) = \sqrt{(x-a)^2 + (y-b)^2}$$

Figure 1.13: Euclidean distance formula

$$manhattanDistance = \sum_{i=1}^{n} \left|p_i - q_i\right|$$

Figure 1.14: Manhattan distance formula

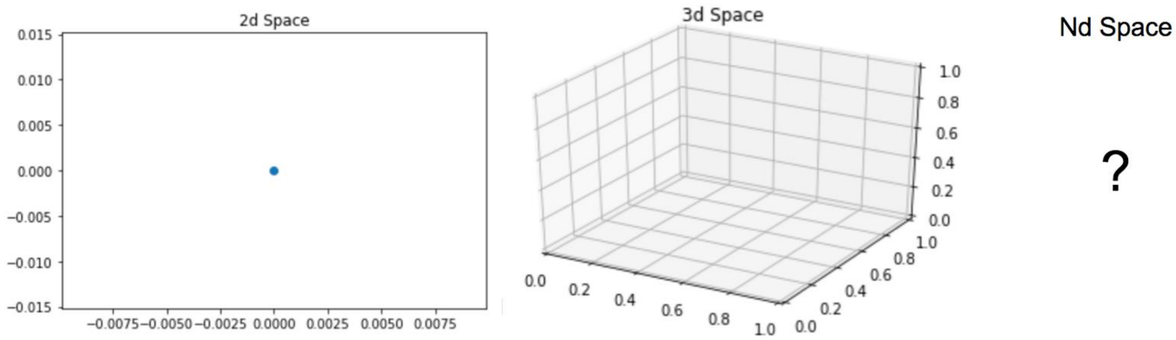


Figure 1.15: Two-dimensional, three-dimensional, and n-dimensional plots

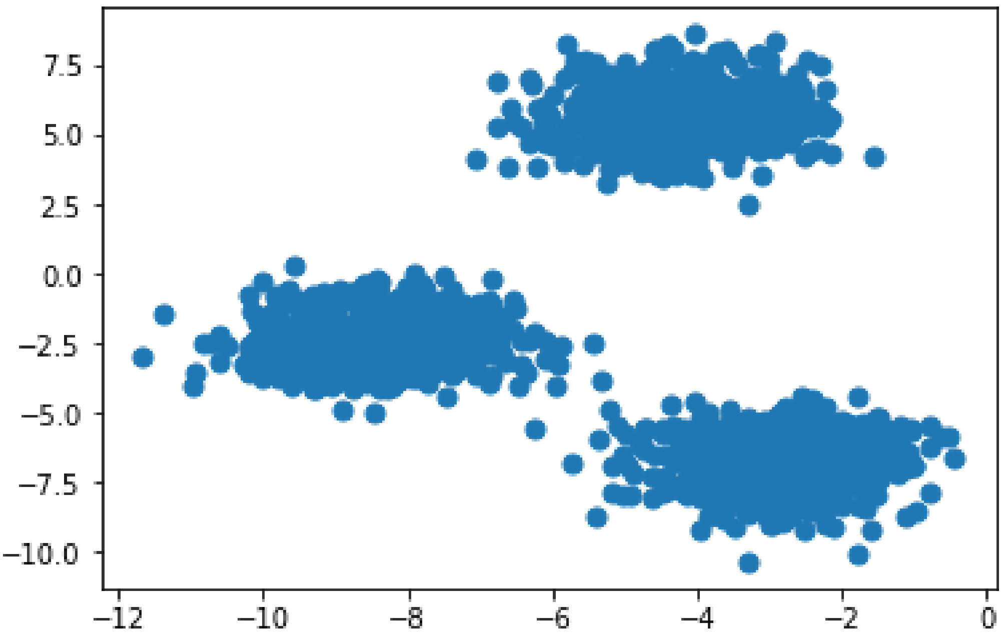


Figure 1.16: Plot of the coordinates

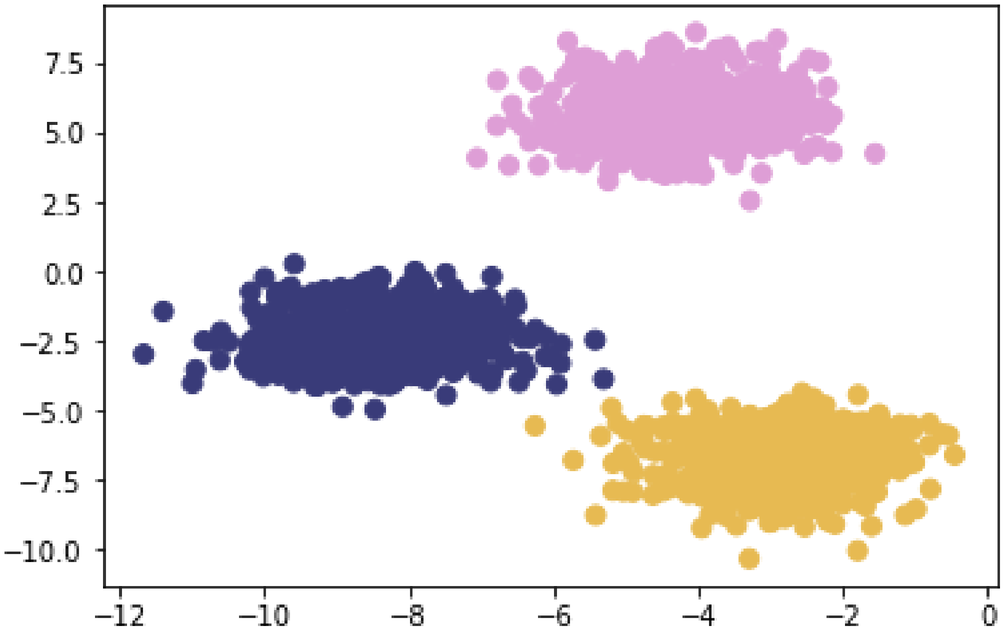


Figure 1.17: Plot of the coordinates with correct cluster labels

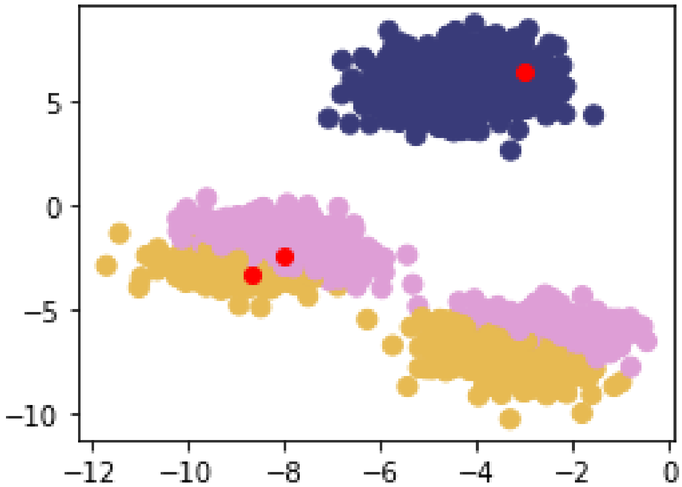


Figure 1.18: First scatterplot

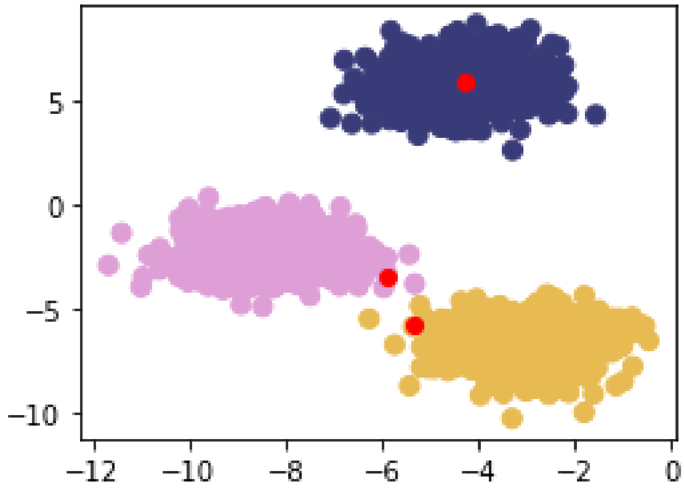


Figure 1.19: Second scatterplot

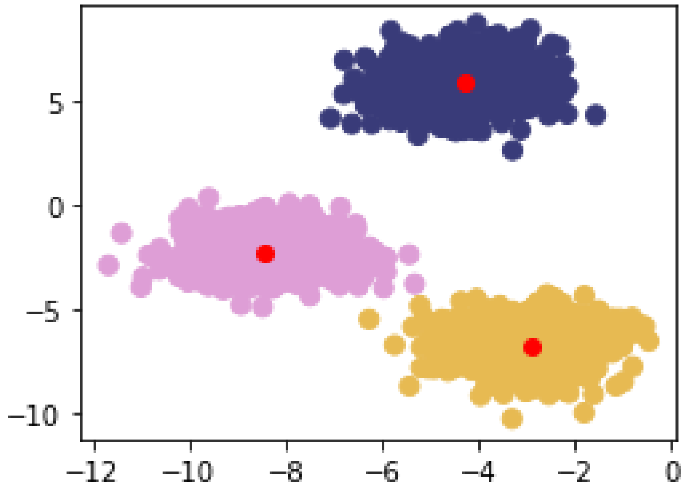


Figure 1.20: Third scatterplot

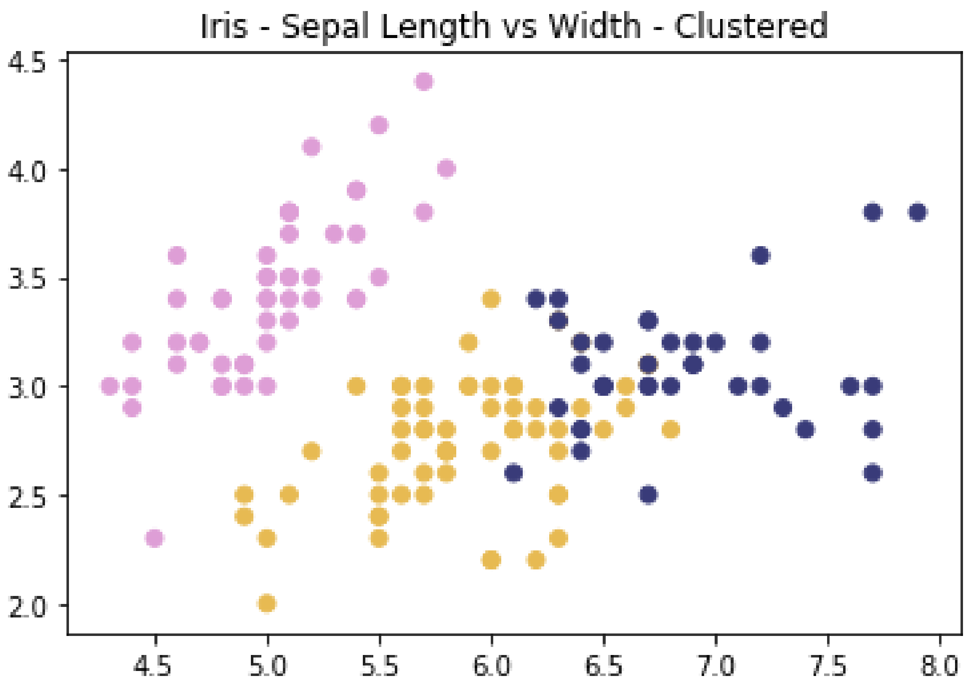


Figure 1.21: Expected plot of three clusters of Iris species

Lesson 2: Hierarchical Clustering

Figure 2.1: The attributes that separate supervised and unsupervised problems

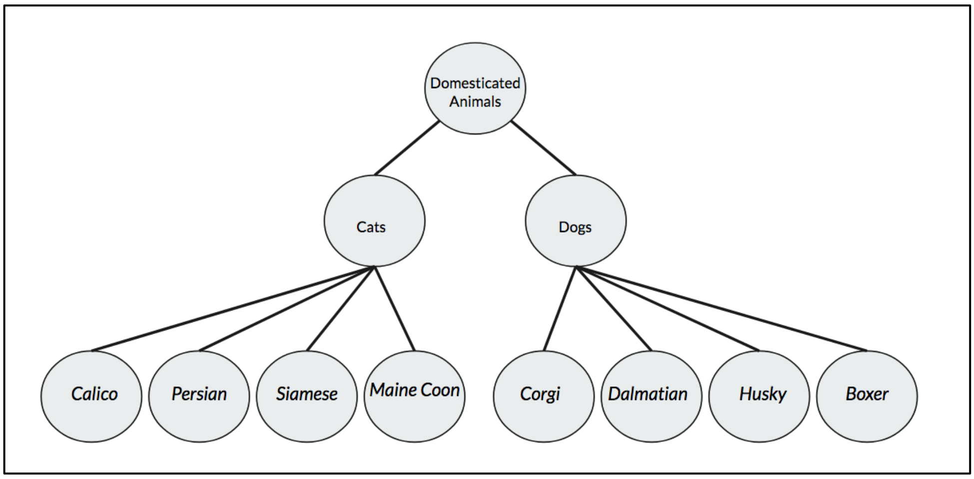


Figure 2.2: Navigating the relationships of animal species in a hierarchical tree structure

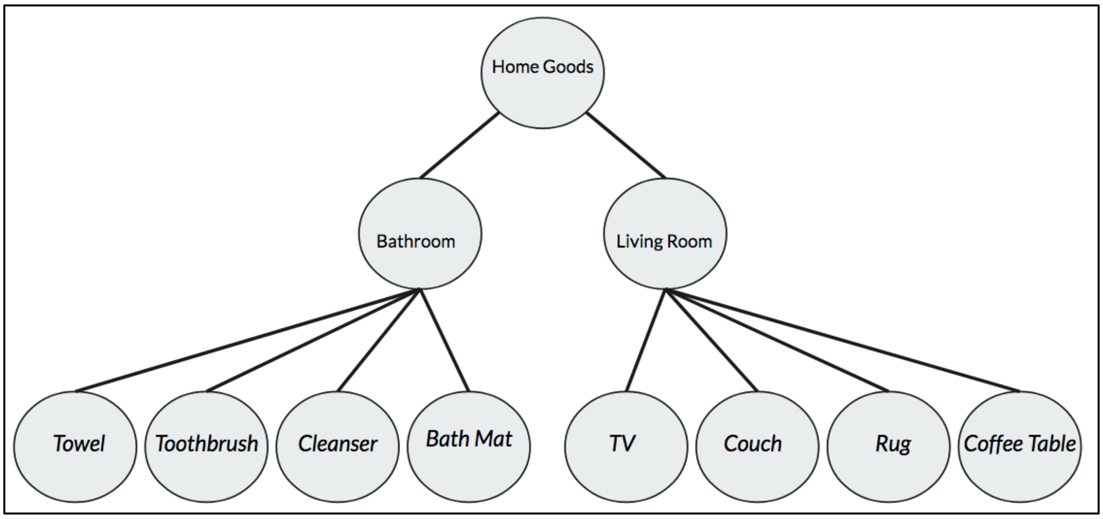


Figure 2.3: Navigating product categories in a hierarchical tree structure

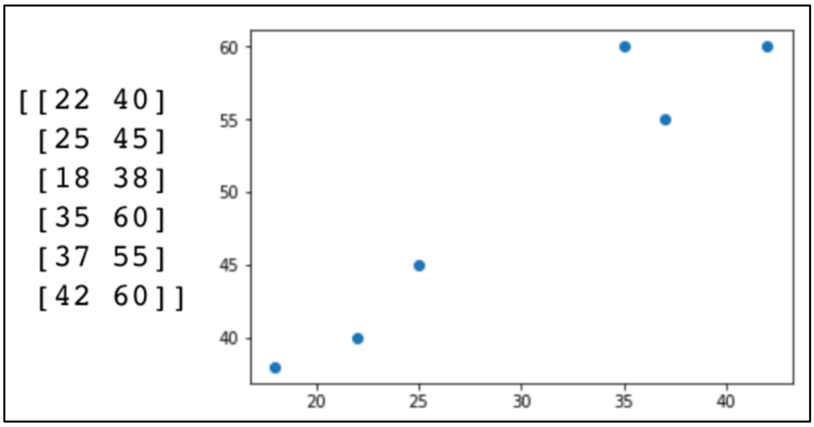


Figure 2.4: An example of a two-feature dataset comprising animal height and animal weight

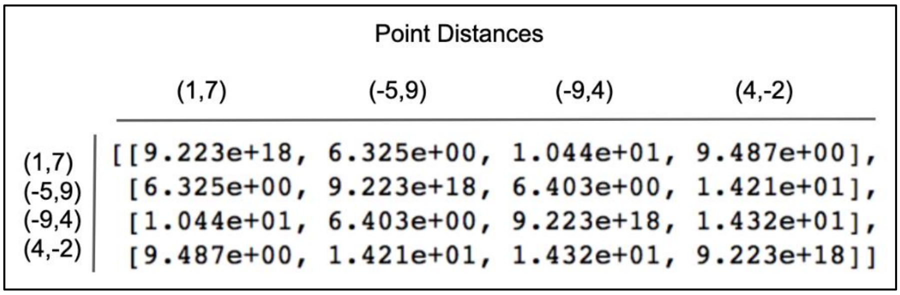


Figure 2.5: An array of distances

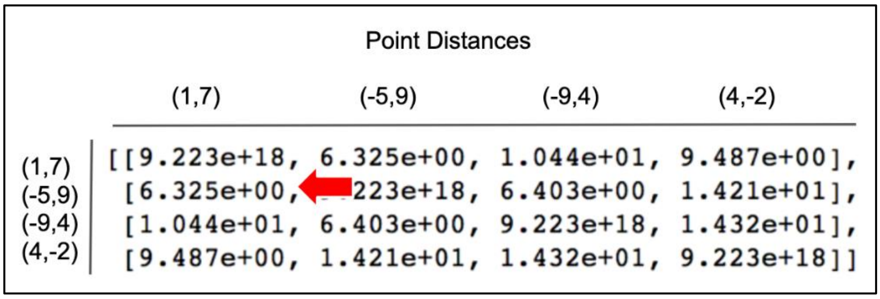


Figure 2.6: An array of distances

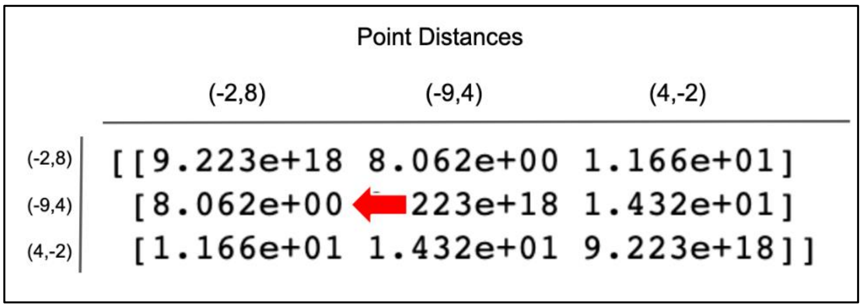


Figure 2.7: An array of distances

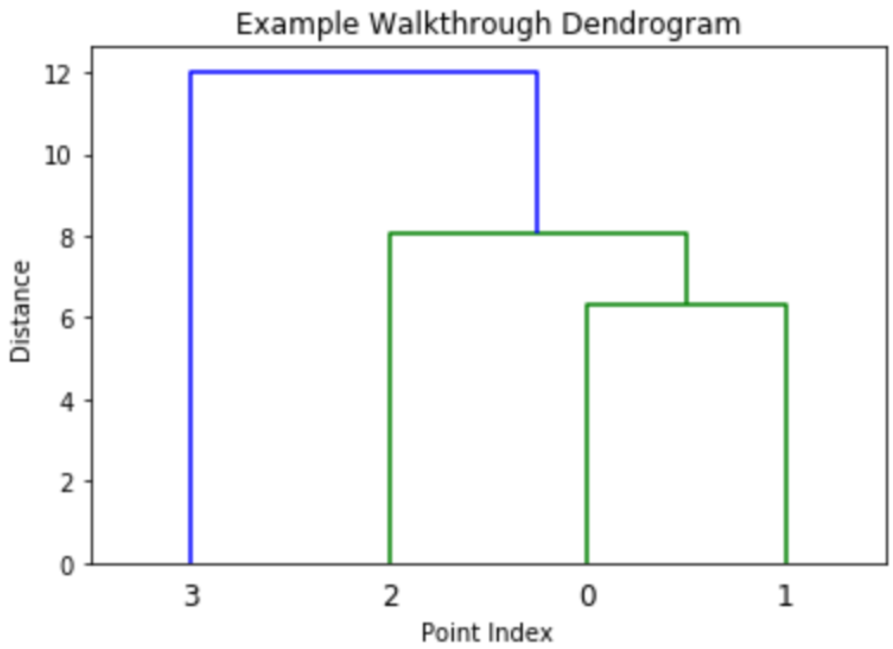


Figure 2.8: A dendrogram showing the relationship between the points and the clusters

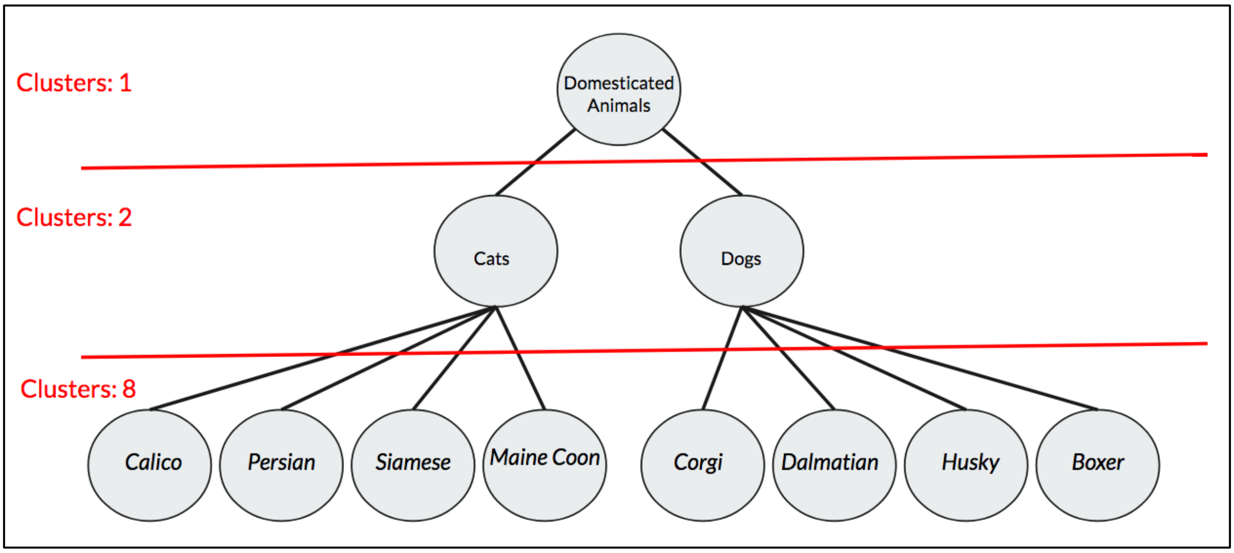


Figure 2.9: An animal taxonomy dendrogram

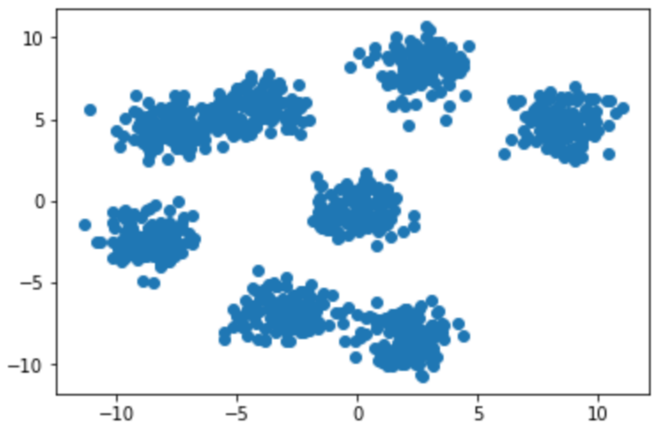


Figure 2.10: A plot of the dummy data

```
[[5.720e+02 7.620e+02 7.694e-03 2.000e+00]
 [3.000e+01 1.960e+02 8.879e-03 2.000e+00]
 [5.910e+02 8.700e+02 1.075e-02 2.000e+00]
 ...
 [1.989e+03 1.992e+03 7.812e+00 3.750e+02]
 [1.995e+03 1.996e+03 1.024e+01 7.500e+02]
 [1.994e+03 1.997e+03 1.200e+01 1.000e+03]]
```

Figure 2.11: A matrix of the distances

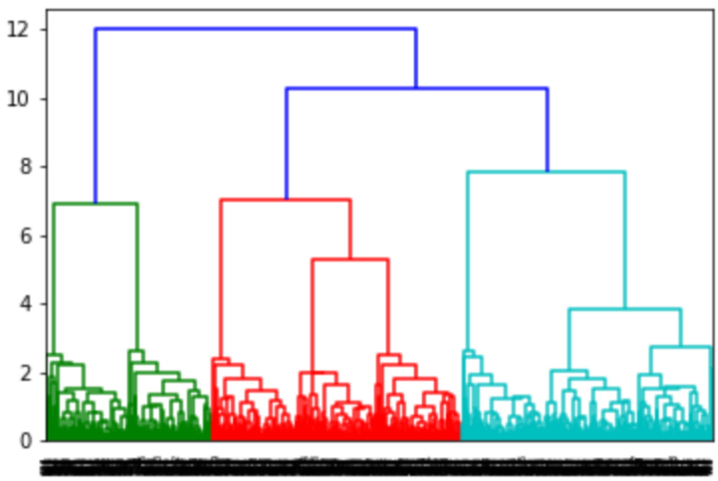


Figure 2.12: A dendrogram of the distances

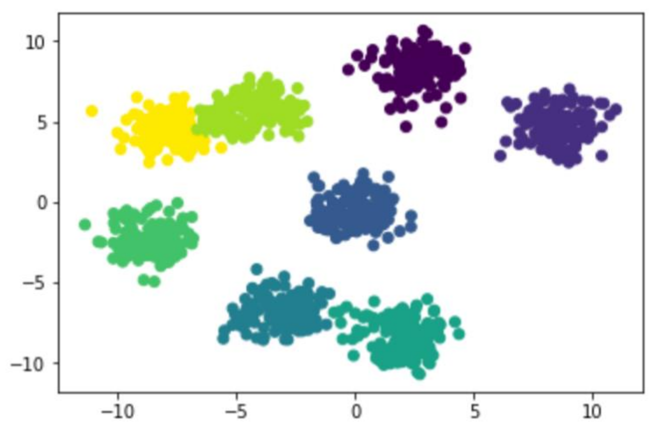


Figure 2.13: A scatter plot of the distances

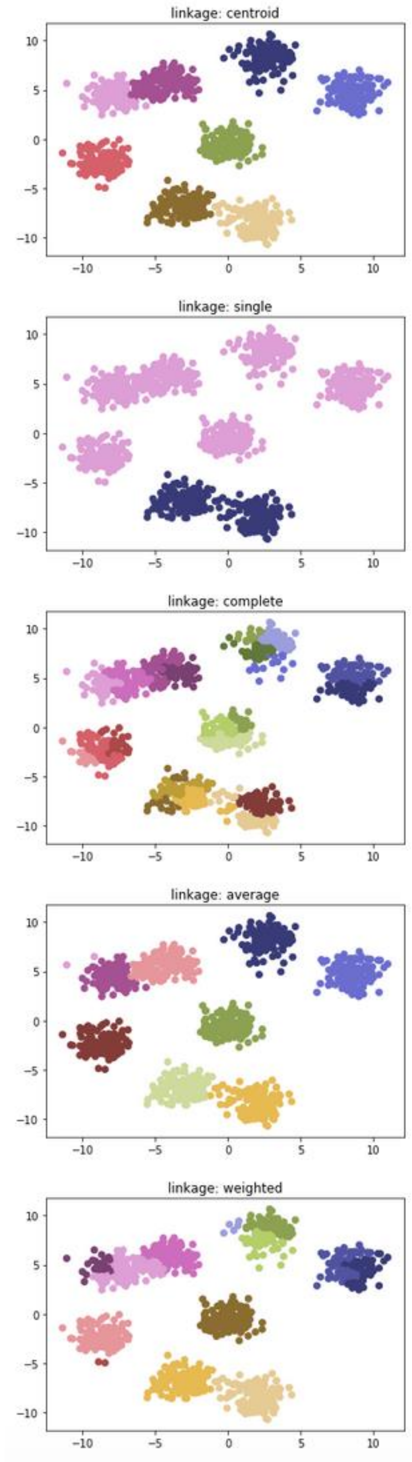


Figure 2.14: The expected scatter plots for all methods

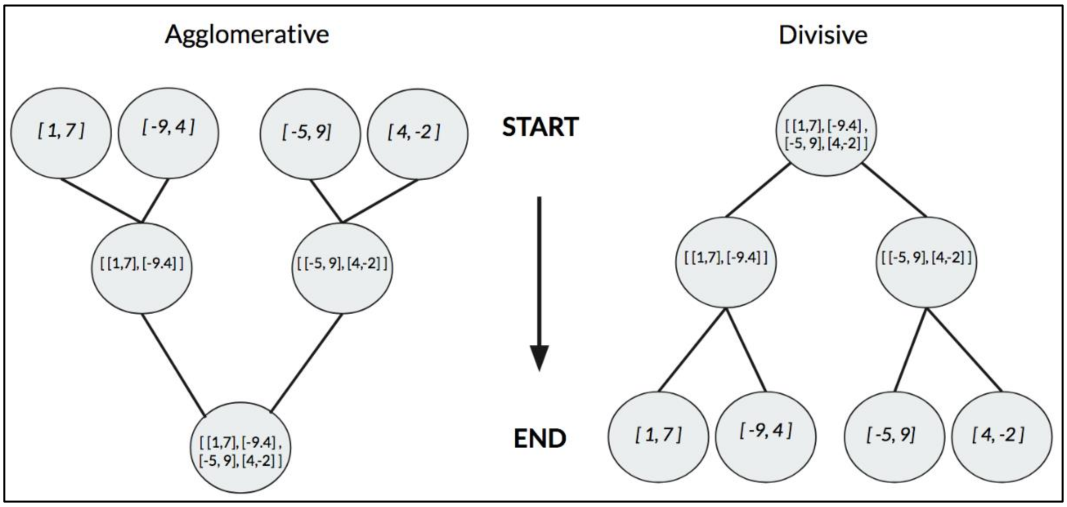


Figure 2.15: Agglomerative versus divisive hierarchical clustering

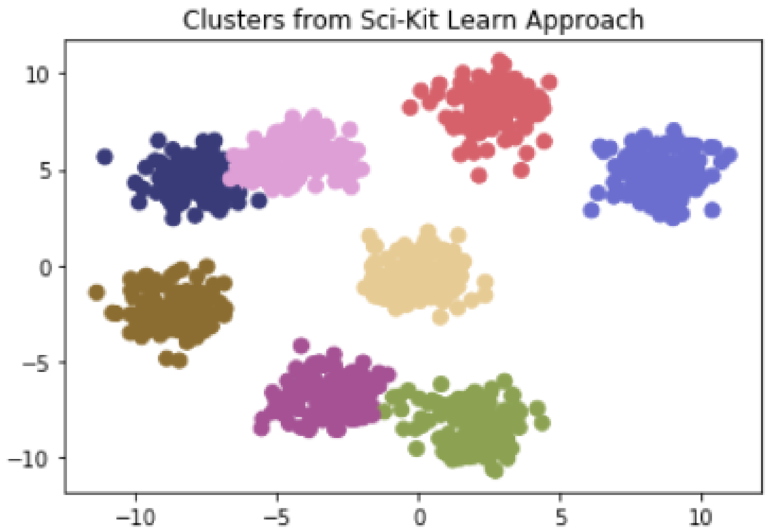


Figure 2.16: A plot of the Scikit-Learn approach

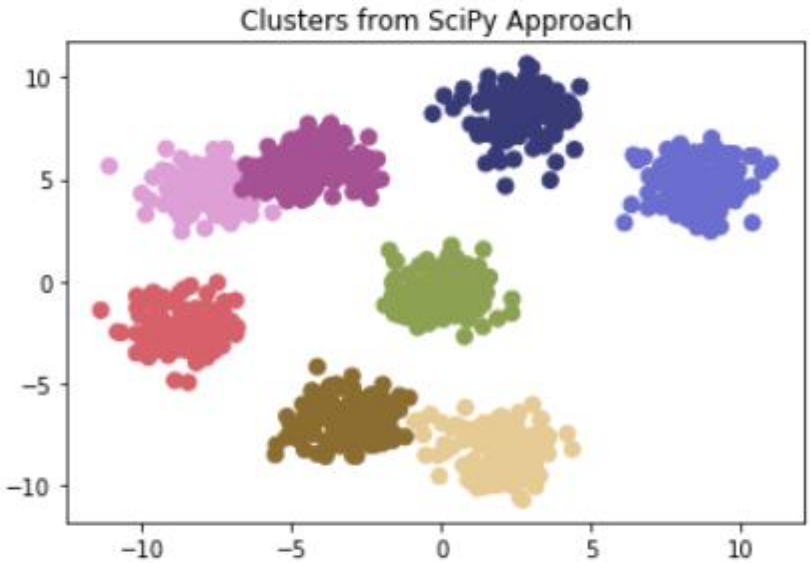


Figure 2.17: A plot of the SciPy approach

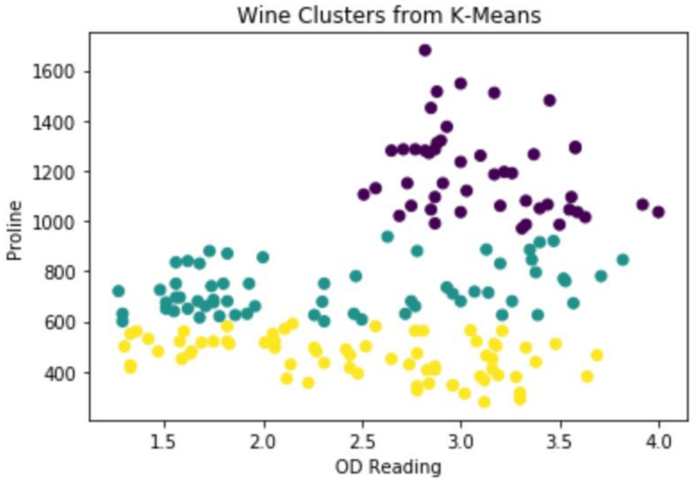


Figure 2.18: The expected clusters from the k-means method

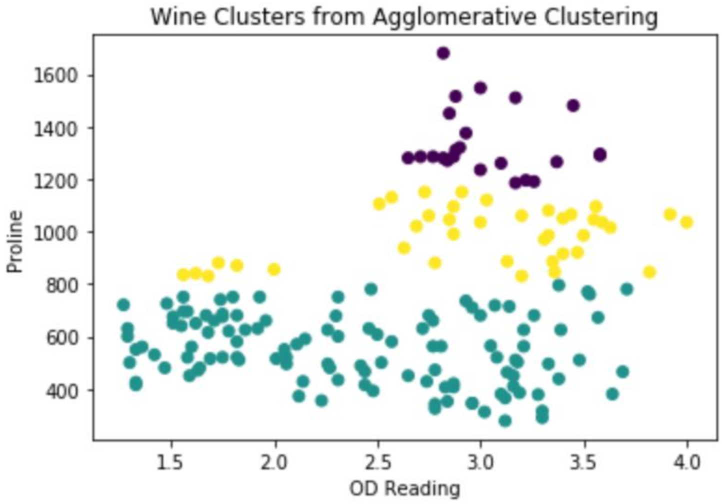


Figure 2.19: The expected clusters from the agglomerative method

Lesson 3: Neighborhood Approaches and DBSCAN

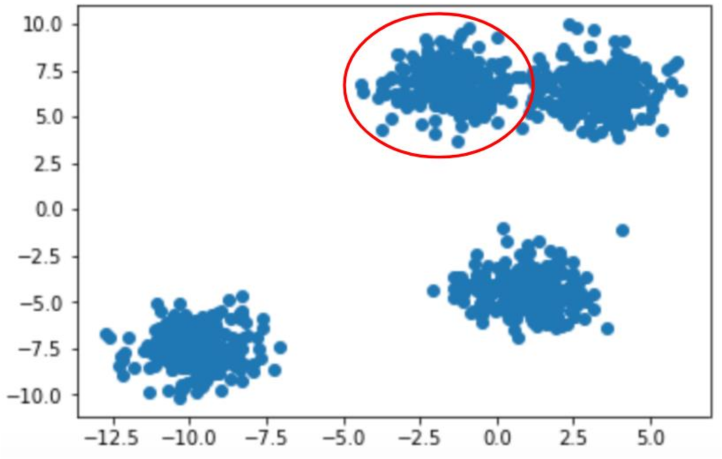


Figure 3.1: Neighbors have a direct connection to clusters

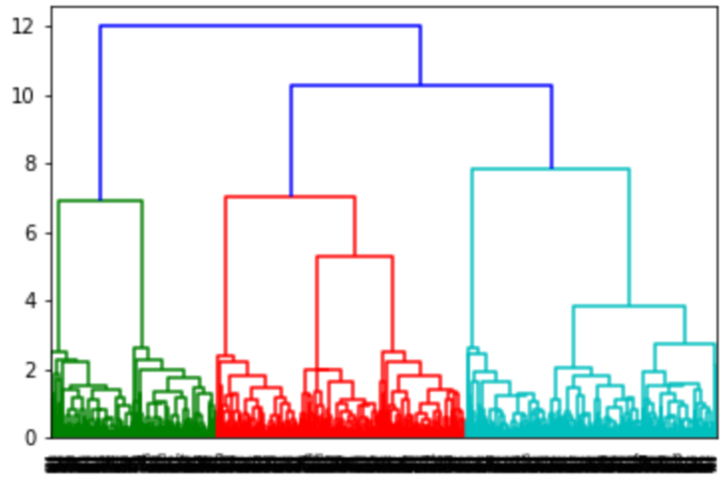


Figure 3.2: Example dendrogram

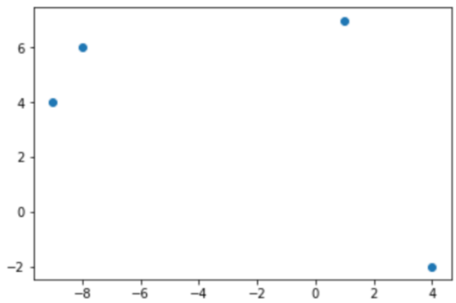


Figure 3.3: Plot of sample data points

Point Distances

	(1,7)	(-5,9)	(-9,4)	(4,-2)
(1,7)	[[9.223e+18,	6.325e+00,	1.044e+01,	9.487e+00],
(-5,9)	[6.325e+00,	9.223e+18,	6.403e+00,	1.421e+01],
(-9,4)	[1.044e+01,	6.403e+00,	9.223e+18,	1.432e+01],
(4,-2)	[9.487e+00,	1.421e+01,	1.432e+01,	9.223e+18]]

Figure 3.4: Point distances

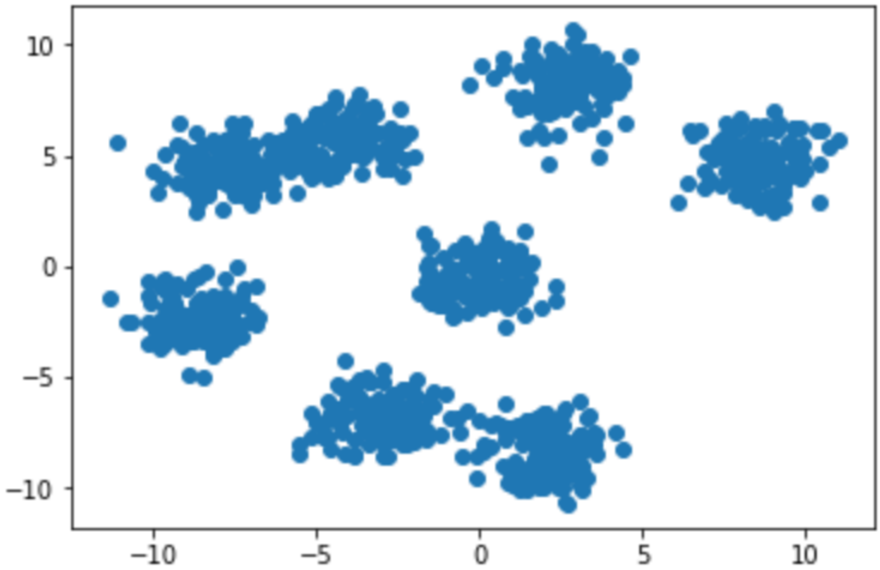


Figure 3.5: Visualized Toy Data Example

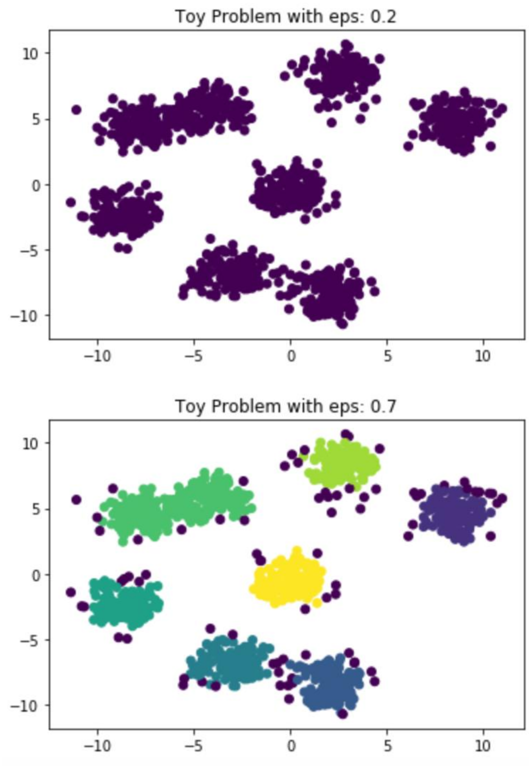


Figure: 3.6: Resulting plots

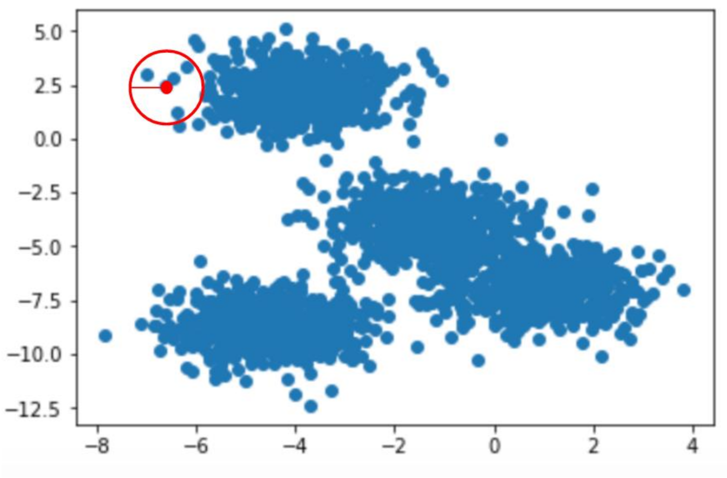


Figure 3.7: Visualization of neighborhood radius where red circle is the neighborhood

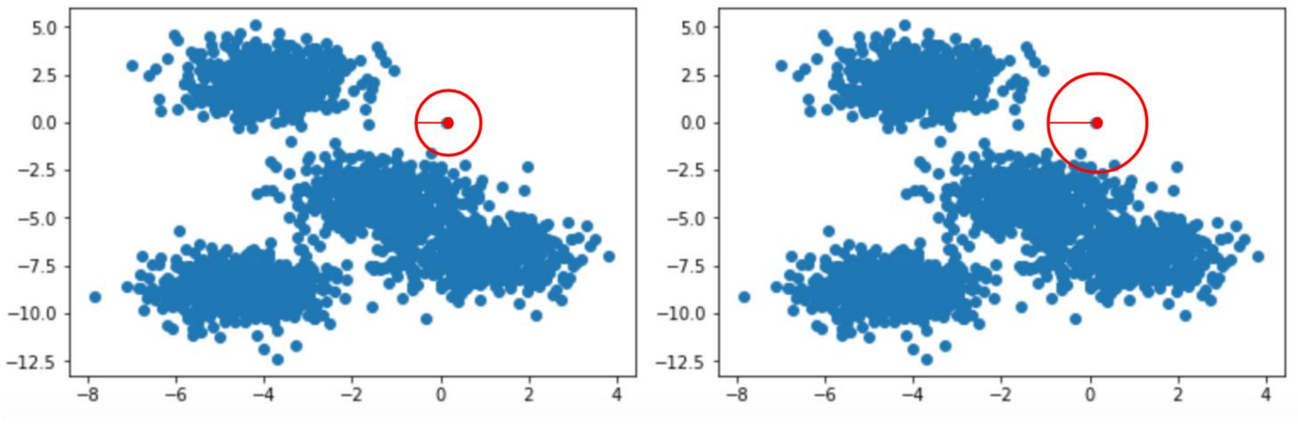


Figure 3.8: Impact of varying neighborhood radius size

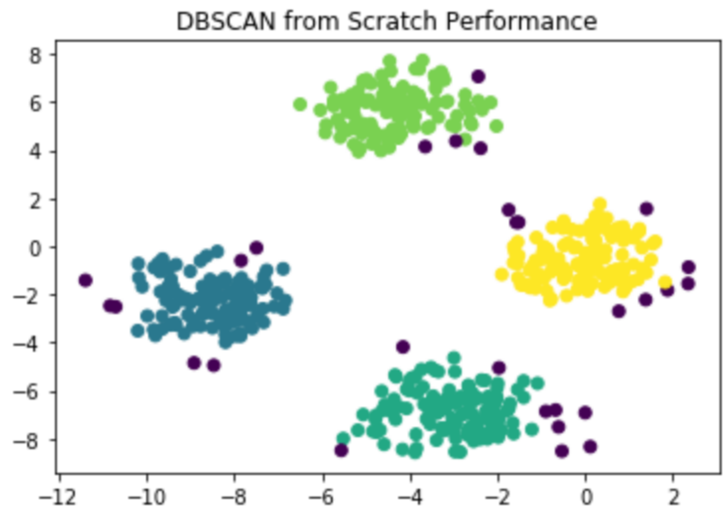


Figure 3.9: Expected outcome

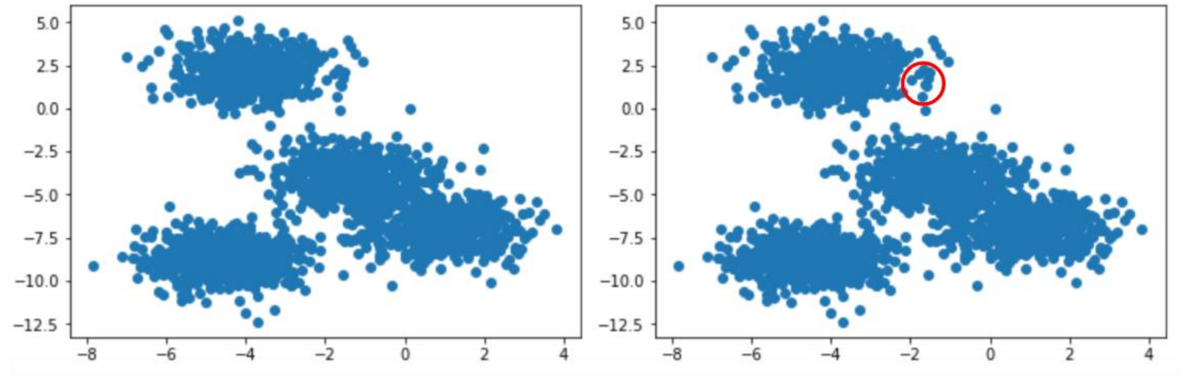


Figure 3.10: Minimum points threshold deciding whether a group of data points is noise or a cluster

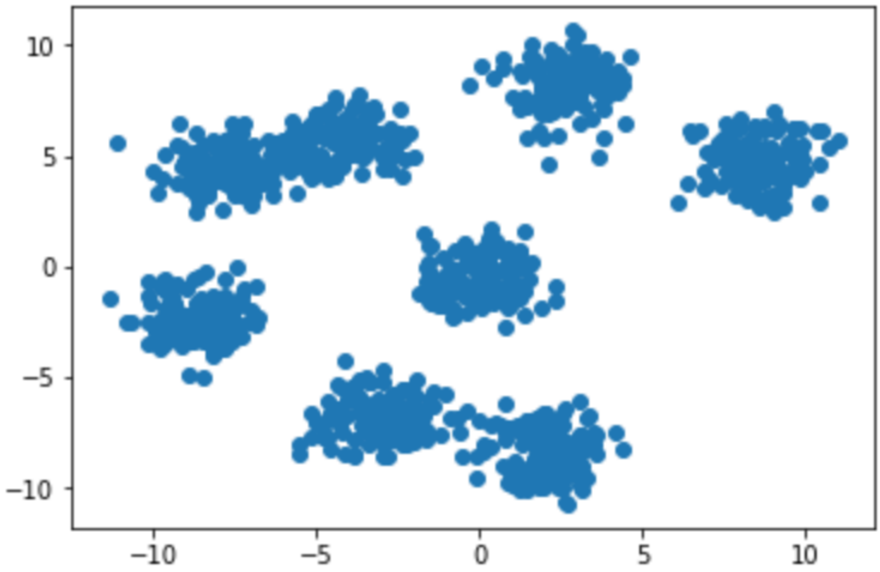


Figure 3.11: Plot of generated data

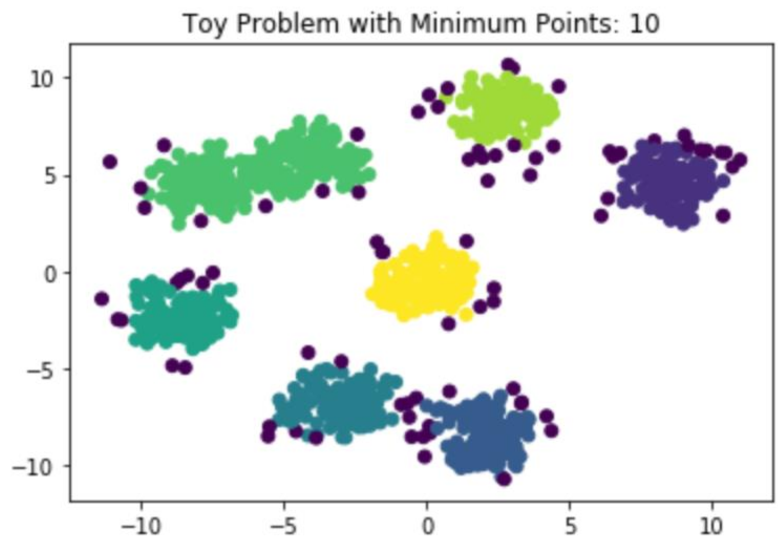


Figure 3.12: Plot of Toy problem with a minimum of 10 points

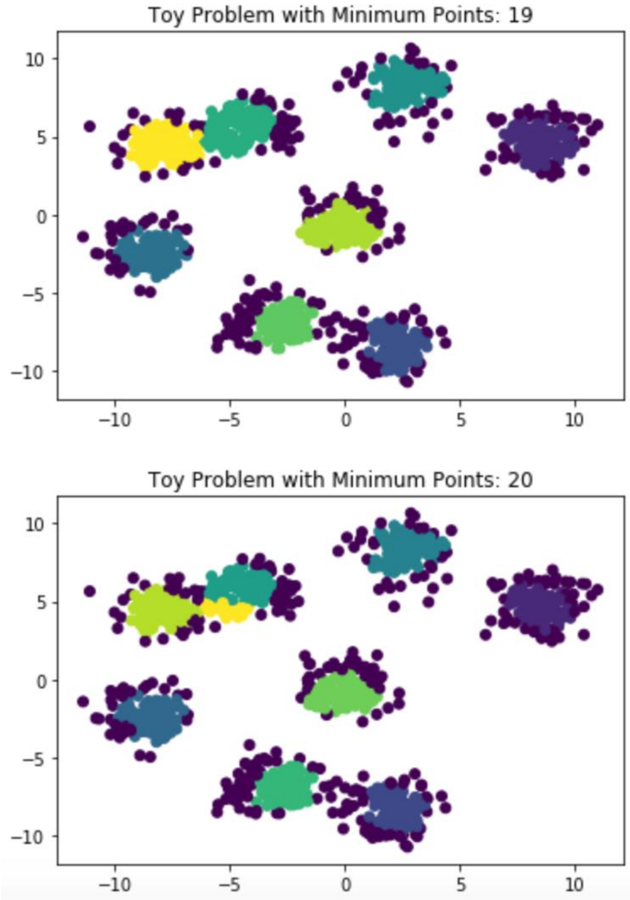


Figure 3.13: Plots of the Toy problem

Lesson 4: An Introduction to Dimensionality Reduction and PCA

Pressure (hPa)	Temperature (^{0}C)	Humidity (%)
1050	32.2	12
1026	27.8	80

Figure 4.1: Two samples of data with three different features

Figure 4.2: Electrocardiogram (ECG or EKG)

Figure 4.3: An image filtered with dimensionality reduction. Left: The original image (Photo by Arthur Brognoli from Pexels), Right: The filtered image

Figure 4.1: Dimensions in a PacMan game

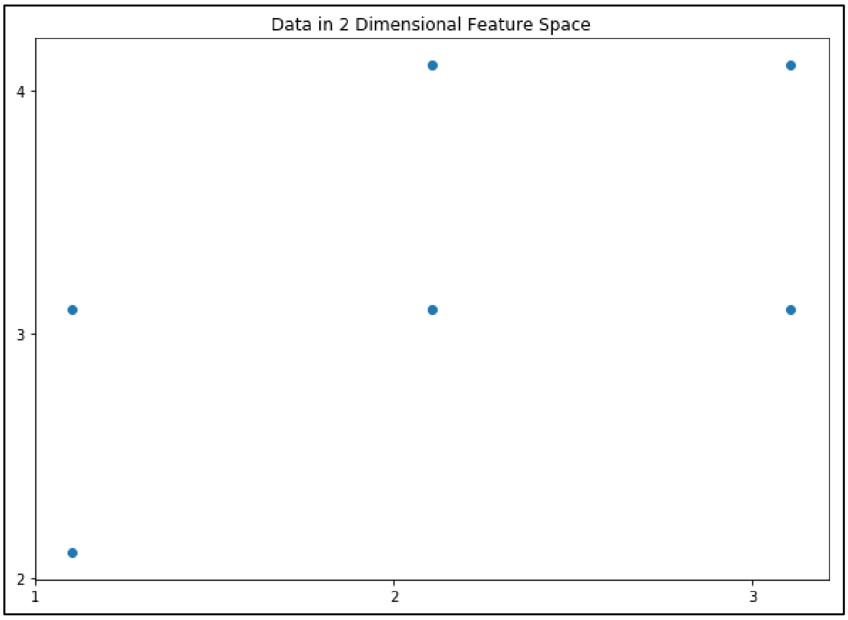


Figure 4.2: Data in a 2D feature space

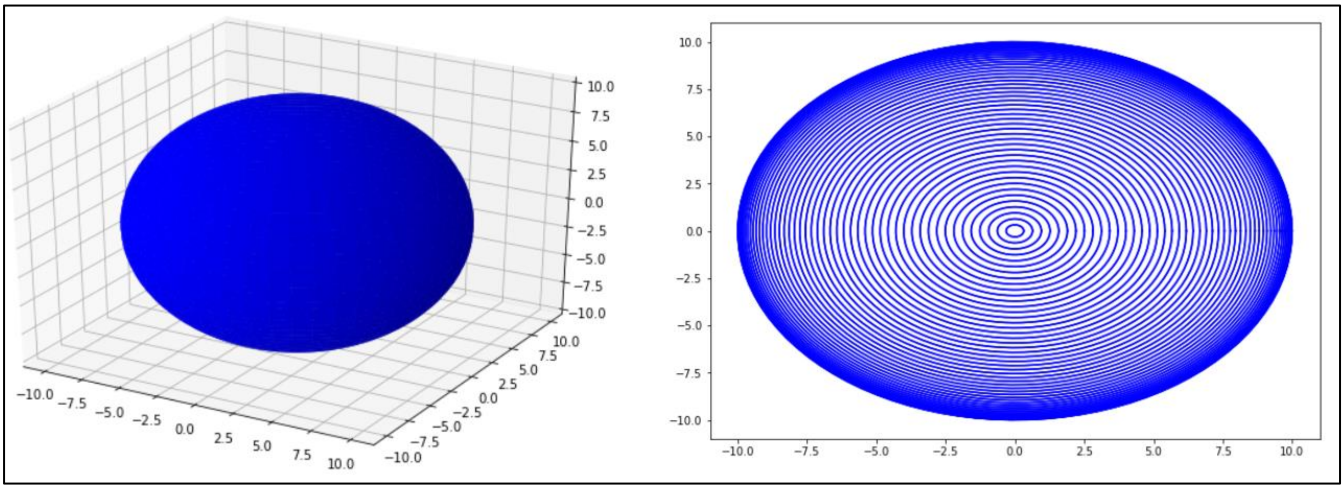


Figure 4.3: A projection of a 3D sphere into a 2D space

$$\overline{cov} = \begin{bmatrix} cov(X,X) & cov(X,Y) & cov(X,Z) \\ cov(Y,X) & cov(Y,Y) & cov(Y,Z) \\ cov(Z,X) & cov(Z,Y) & cov(Z,Z) \end{bmatrix}$$

Figure 4.7: The covariance matrix

	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Figure 4.8: The head of the data

	Sepal Length	Sepal Width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6

Figure 4.9: The head after cleaning the data

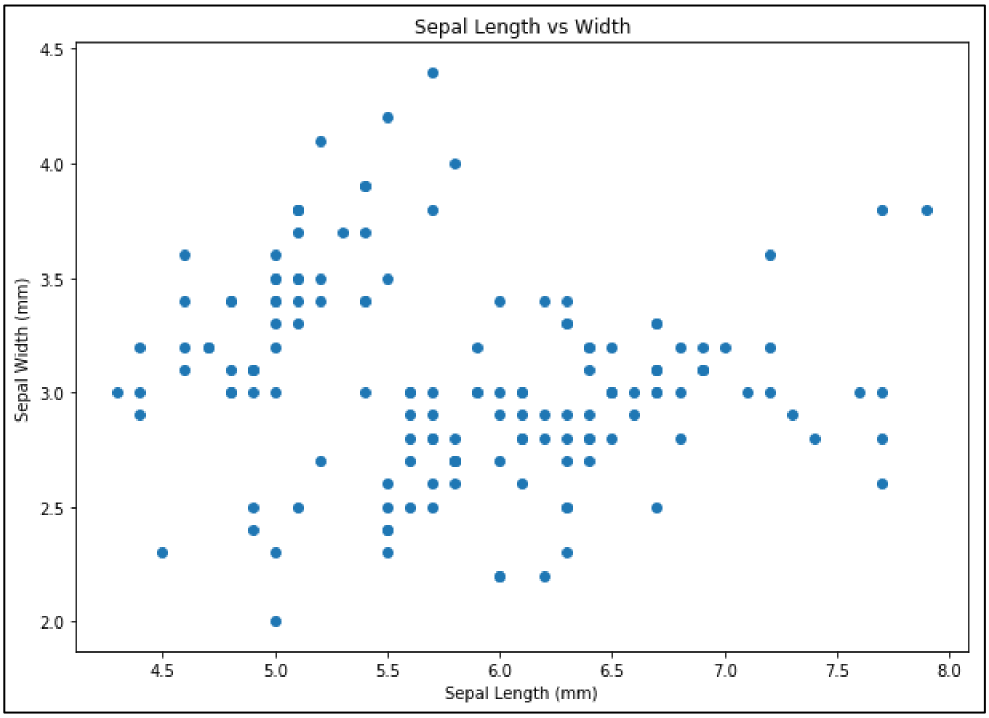


Figure 4.10: Plot of the data

	Sepal Length	Sepal Width
Sepal Length	0.685694	-0.039268
Sepal Width	-0.039268	0.188004

Figure 4.11: Covariance matrix using the Pandas method

```
array([[ 0.68569351, -0.03926846],
       [-0.03926846,  0.18800403]])
```

Figure 4.12: The covariance matrix using the NumPy method

$$a = USV^T$$

Figure 4.13: An eigenvector/eigenvalue decomposition

	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Figure 4.14: The first five rows of the dataset

	Sepal Length	Sepal Width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6

Figure 4.15: The Sepal Length and Sepal Width feature

```
array([[-0.07553027, -0.11068158],
       [-0.07052087, -0.06007995],
       [-0.06946245, -0.09874988],
       [-0.06780439, -0.09257869],
       [-0.07500106, -0.13001654],
       [-0.08106887, -0.14194824],
       [-0.06949767, -0.13083793],
       [-0.07387221, -0.10451038],
```

Figure 4.16: Eigenvectors

	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Figure 4.17: The first five rows of the dataset

	Sepal Length	Sepal Width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6

Figure 4.18: The sepal length and sepal width feature

```
array([[ 0.68569351, -0.03926846],
       [-0.03926846,  0.18800403]])
```

Figure 4.19: The covariance matrix for the selected data

```
array([[-0.99693955,  0.07817635],
       [ 0.07817635,  0.99693955]])
```

Figure 4.20: Eigenvectors

```
array([-4.81077444, -4.65047471, -4.43545153, -4.34357521, -4.70326285,
       -5.07858577, -4.32012231, -4.71889812, -4.15982257, -4.64265708,
       -5.09422104, -4.51951021, -4.55078076, -4.05231098, -5.46954395,
       -5.33857945, -5.07858577, -4.81077444, -5.38548526, -4.78732154,
       -5.11767394, -4.79513917, -4.30448703, -4.82640971, -4.51951021,
       -4.75016867, -4.71889812, -4.9104684 , -4.91828603, -4.43545153,
       -4.54296312, -5.11767394, -4.86356259, -5.15482681, -4.64265708,
       -4.73453339, -5.20955026, -4.64265708, -4.15200494, -4.81859208,
       -4.71108049, -4.30642234, -4.13636967, -4.71108049, -4.78732154,
```

Figure 4.21: The result of matrix multiplication

```
array([-4.81077444, -4.65047471, -4.43545153, -4.34357521, -4.70326285,
       -5.07858577, -4.32012231, -4.71889812, -4.15982257, -4.64265708,
       -5.09422104, -4.51951021, -4.55078076, -4.05231098, -5.46954395,
       -5.33857945, -5.07858577, -4.81077444, -5.38548526, -4.78732154,
       -5.11767394, -4.79513917, -4.30448703, -4.82640971, -4.51951021,
       -4.75016867, -4.71889812, -4.9104684 , -4.91828603, -4.43545153,
       -4.54296312, -5.11767394, -4.86356259, -5.15482681, -4.64265708,
       -4.73453339, -5.20955026, -4.64265708, -4.15200494, -4.81859208,
       -4.71108049, -4.30642234, -4.13636967, -4.71108049, -4.78732154,
       -4.55078076, -4.78732154, -4.33575758, -4.99452708, -4.72671576,
       -6.72841249, -6.13024876, -6.63653617, -5.30336189, -6.26121325,
       -5.46366162, -6.02273717, -4.69738052, -6.35308957, -4.97300948,
       -4.82834502, -5.64741426, -5.80964929, -5.8546198 , -5.35615003,
       -6.43714826, -5.34833239, -5.57117321, -6.0090372 , -5.38742057,
       -5.63177899, -5.86243744, -6.08527825, -5.86243744, -6.15370166,
       -6.34527194, -6.56029512, -6.44496589, -5.75492585, -5.47929689,
```

Figure 4.22: The output of PCA

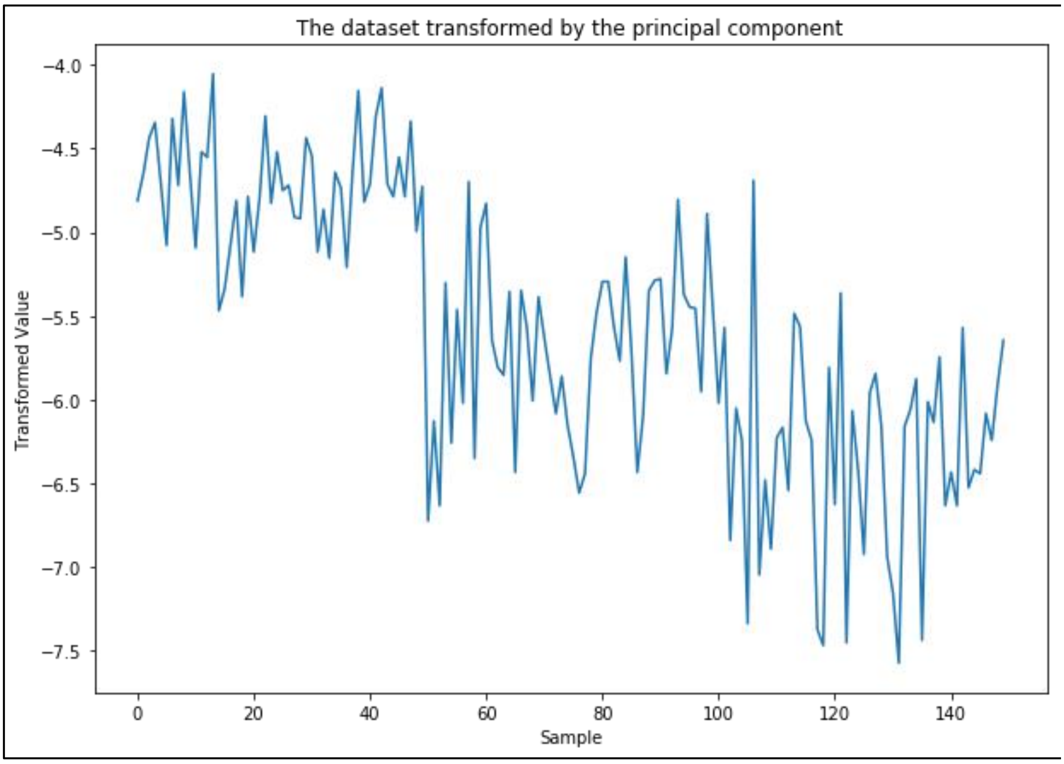


Figure 4.23: The Iris dataset transformed by using a manual PCA

	Sepal Length	Sepal Width	Petal Length	Petal Width	Species
0	5.1	3.5	1.4	0.2	Iris-setosa
1	4.9	3.0	1.4	0.2	Iris-setosa
2	4.7	3.2	1.3	0.2	Iris-setosa
3	4.6	3.1	1.5	0.2	Iris-setosa
4	5.0	3.6	1.4	0.2	Iris-setosa

Figure 4.24: The first five rows of the dataset

	Sepal Length	Sepal Width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6

Figure 4.25: The Sepal Length and Sepal Width features

```
PCA(copy=True, iterated_power='auto', n_components=None, random_state=None,
  svd_solver='auto', tol=0.0, whiten=False)
```

Figure 4.26: Fitting data to a PCA model

```
array([[ 0.99693955, -0.07817635],
       [ 0.07817635,  0.99693955]])
```

Figure 4.27: Eigenvectors

```
PCA(copy=True, iterated_power='auto', n_components=1, random_state=None,
  svd_solver='auto', tol=0.0, whiten=False)
```

Figure 4.28: The maximum number of eigenvalues and eigenvectors

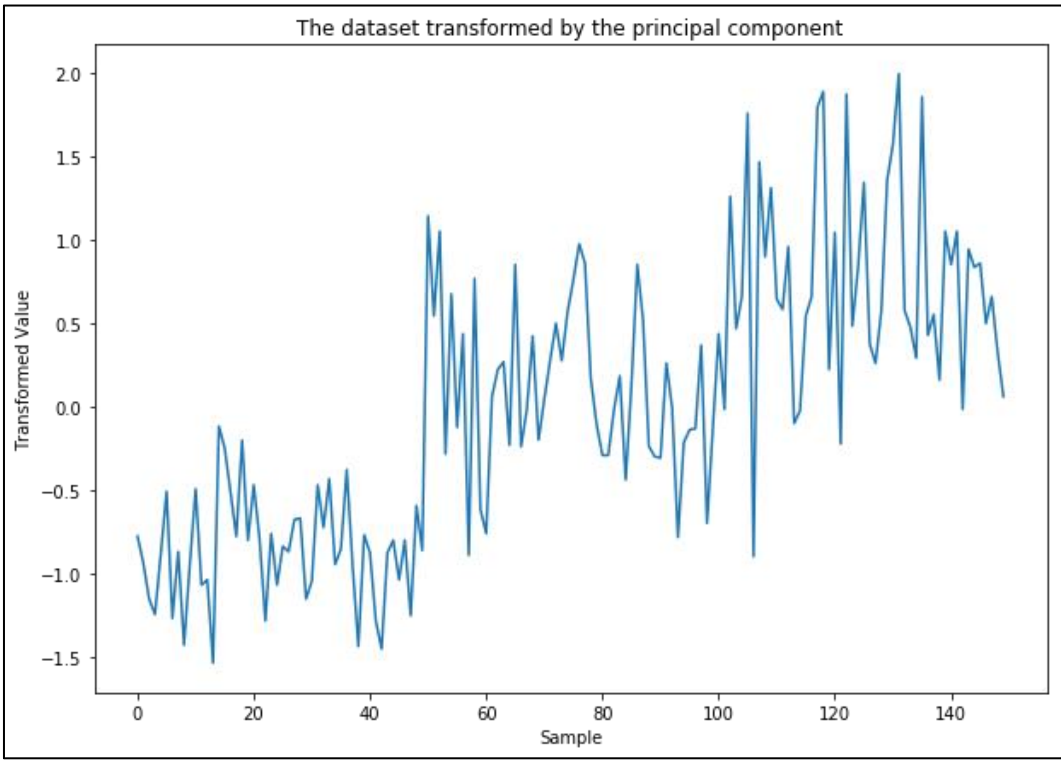


Figure 4.29: The Iris dataset transformed using the scikit-learn PCA

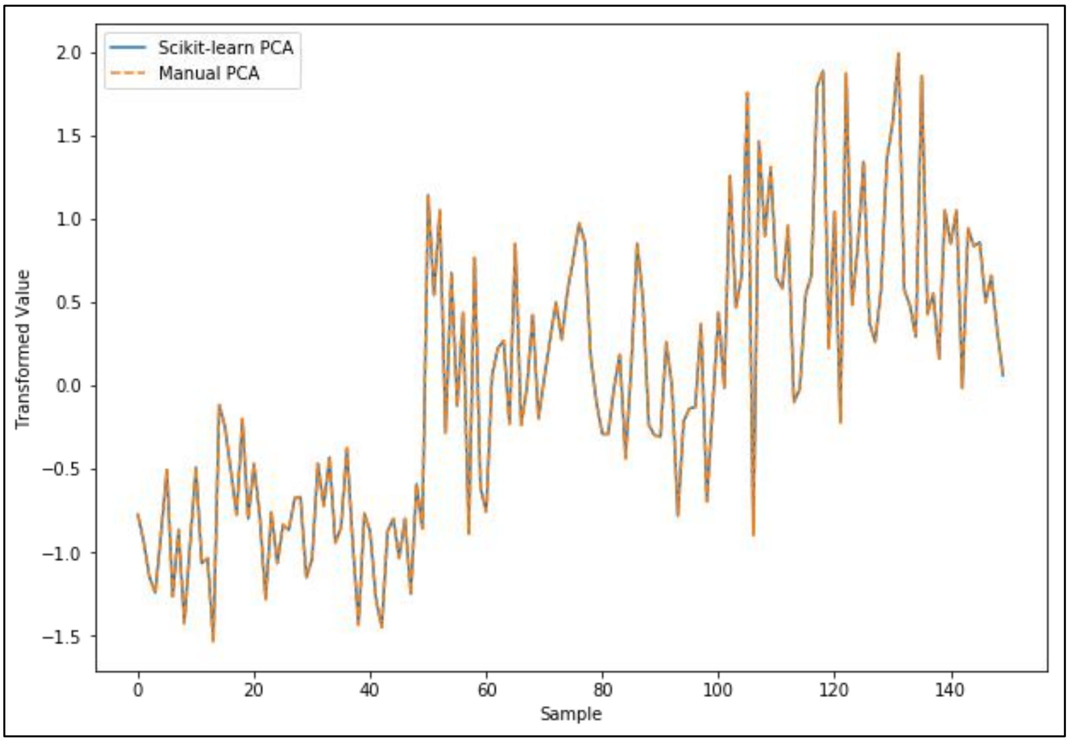


Figure 4.30: The expected final plot

	Sepal Length	Sepal Width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6

Figure 4.31: Sepal features

```
array([[-0.74333333,  0.446     ],
       [-0.94333333, -0.054     ],
       [-1.14333333,  0.146     ],
       [-1.24333333,  0.046     ],
       [-0.84333333,  0.546     ],
       [-0.44333333,  0.846     ],
       [-1.24333333,  0.346     ],
       [-0.84333333,  0.346     ],
```

Figure 4.32: Section of the output

```
array([[-7.73550366e-01,  6.06589915e-02],
       [-9.33359508e-01,  7.31906401e-02],
       [-1.14772462e+00,  9.00003684e-02],
       [-1.23931976e+00,  9.71829241e-02],
       [-8.80732922e-01,  6.90638556e-02],
       [-5.06558669e-01,  3.97224787e-02],
       [-1.26270089e+00,  9.90163868e-02],
       [-8.65145502e-01,  6.78415472e-02],
       [-1.42251003e+00,  1.11548035e-01],
       [-9.41153218e-01,  7.38017944e-02],
```

Figure 4.33: The inverse transform of the reduced data

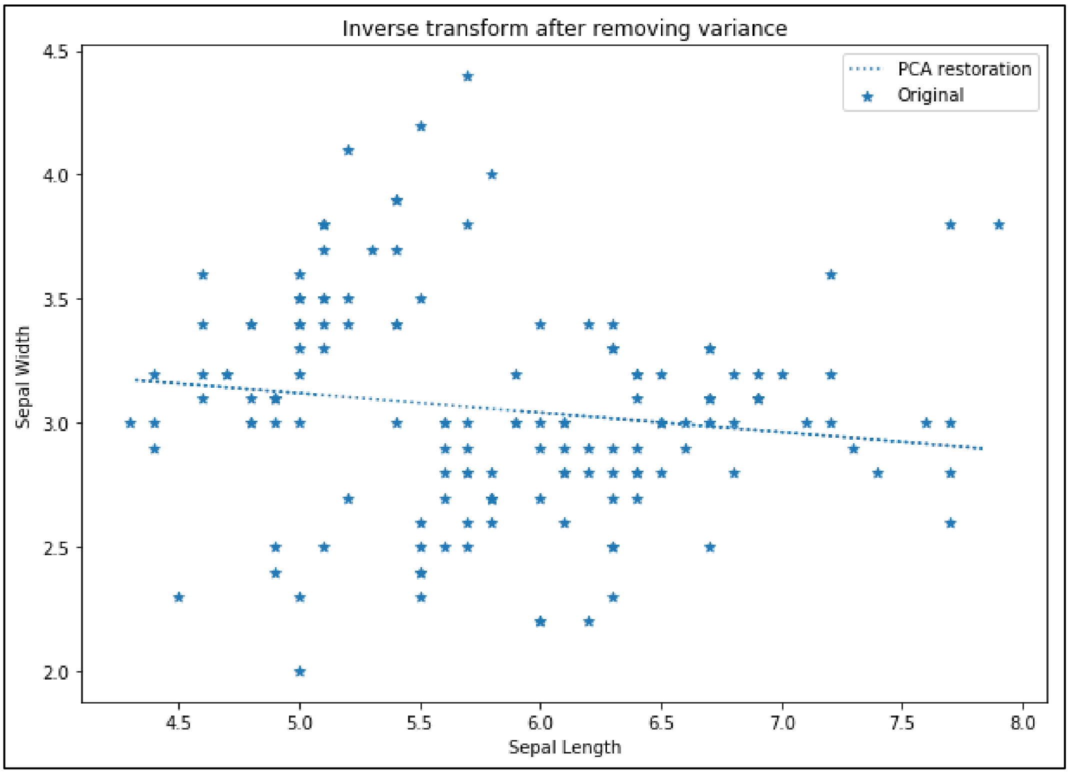


Figure 4.34: The inverse transform after removing variance

```
array([[-0.74333333,  0.446     ],
       [-0.94333333, -0.054     ],
       [-1.14333333,  0.146     ],
       [-1.24333333,  0.046     ],
       [-0.84333333,  0.546     ],
       [-0.44333333,  0.846     ],
       [-1.24333333,  0.346     ],
       [-0.84333333,  0.346     ],
```

Figure 4.35: The restored data

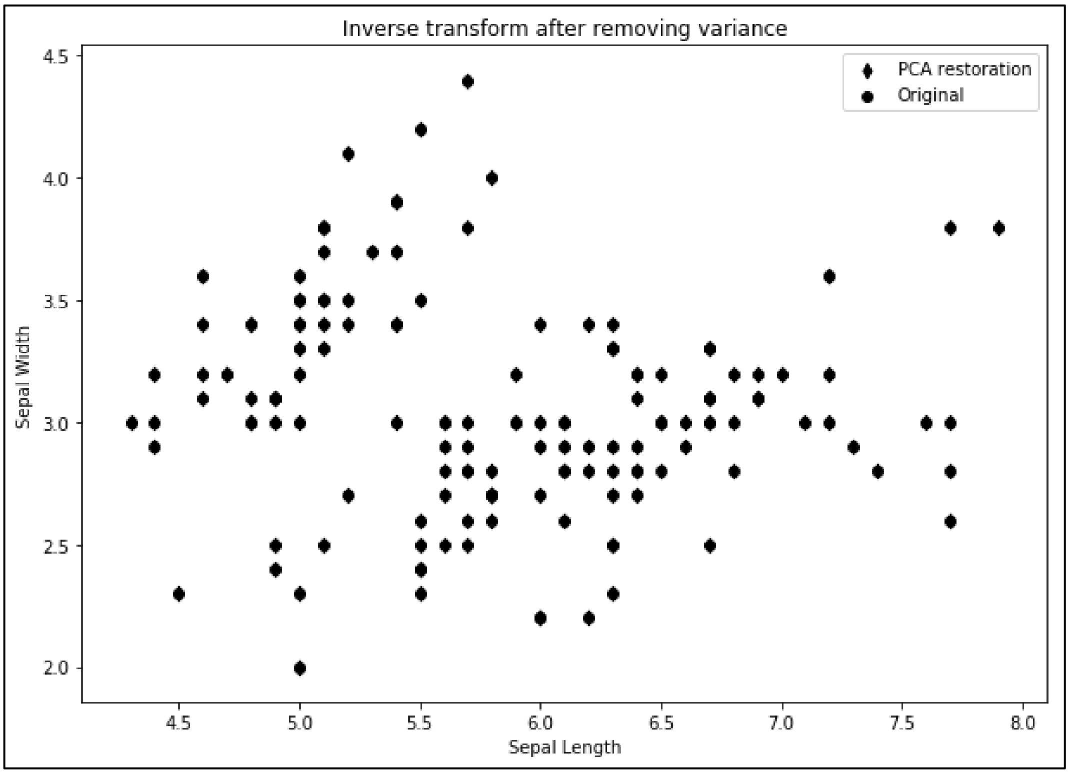


Figure 4.36: The inverse transform after removing the variance

	Sepal Length	Sepal Width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6

Figure 4.37: The Sepal features from the Iris dataset

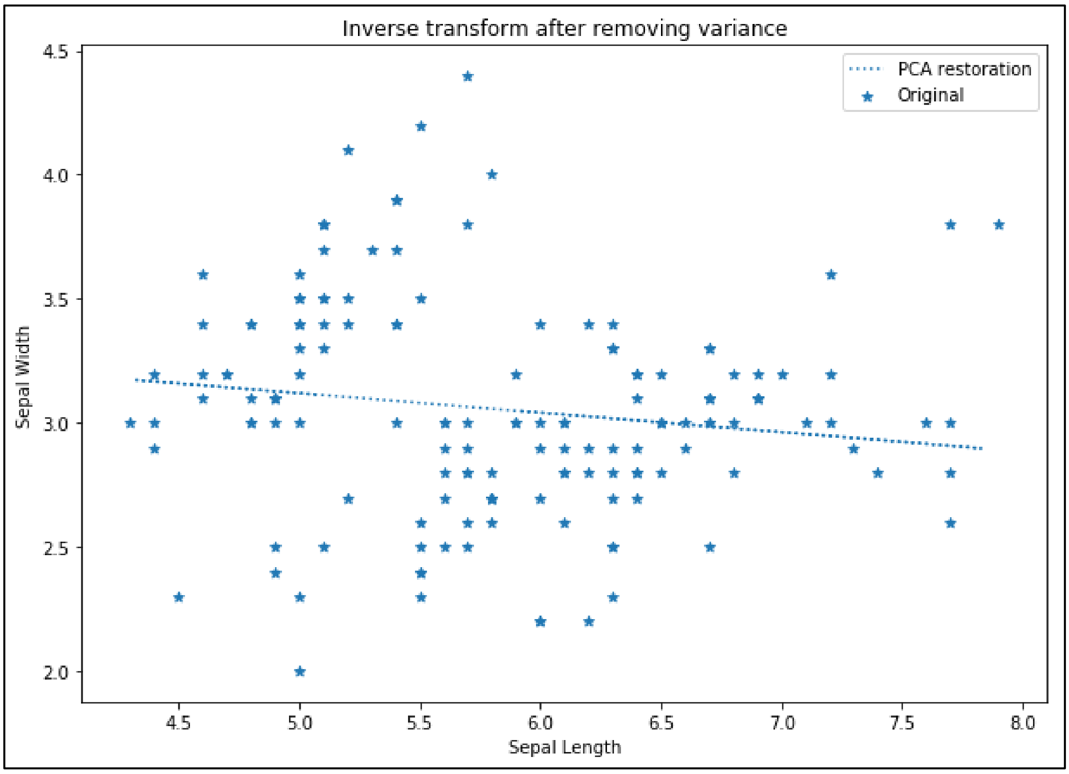


Figure 4.38: The inverse transform after removing the variance

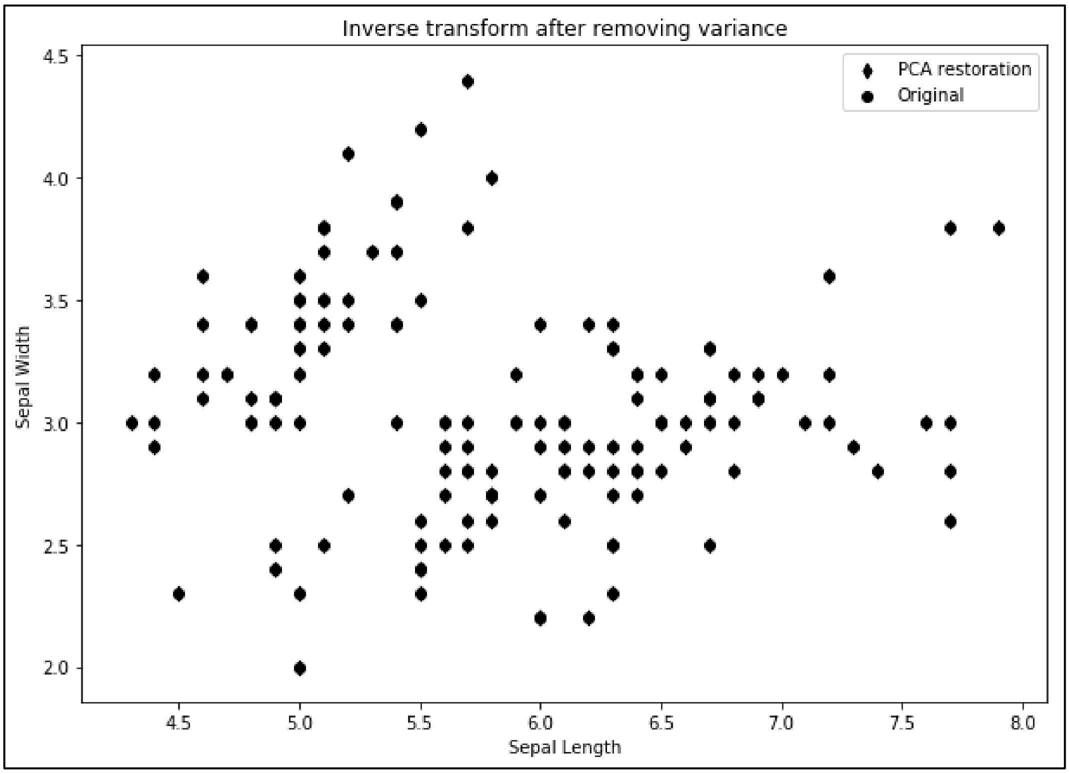


Figure 4.39: The inverse transform after removing the variance

	Sepal Length	Sepal Width	Petal Width
0	5.1	3.5	0.2
1	4.9	3.0	0.2
2	4.7	3.2	0.2
3	4.6	3.1	0.2
4	5.0	3.6	0.2

Figure 4.40: The first five rows of the data

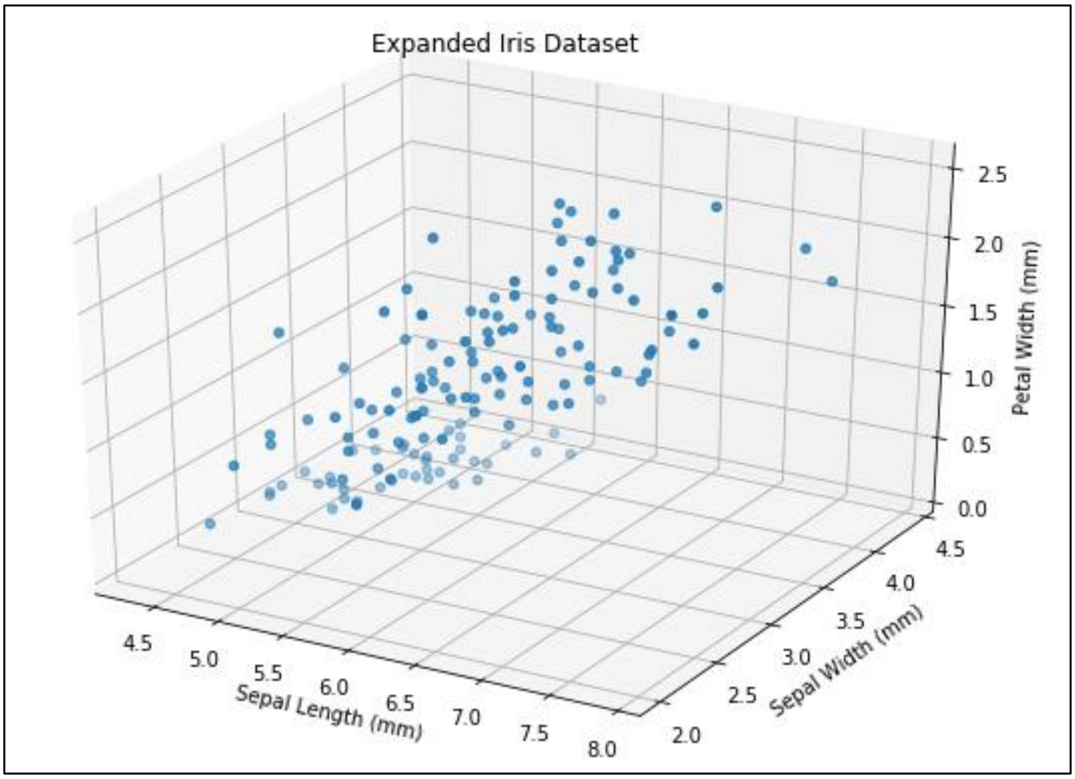


Figure 4.41: The expanded Iris dataset

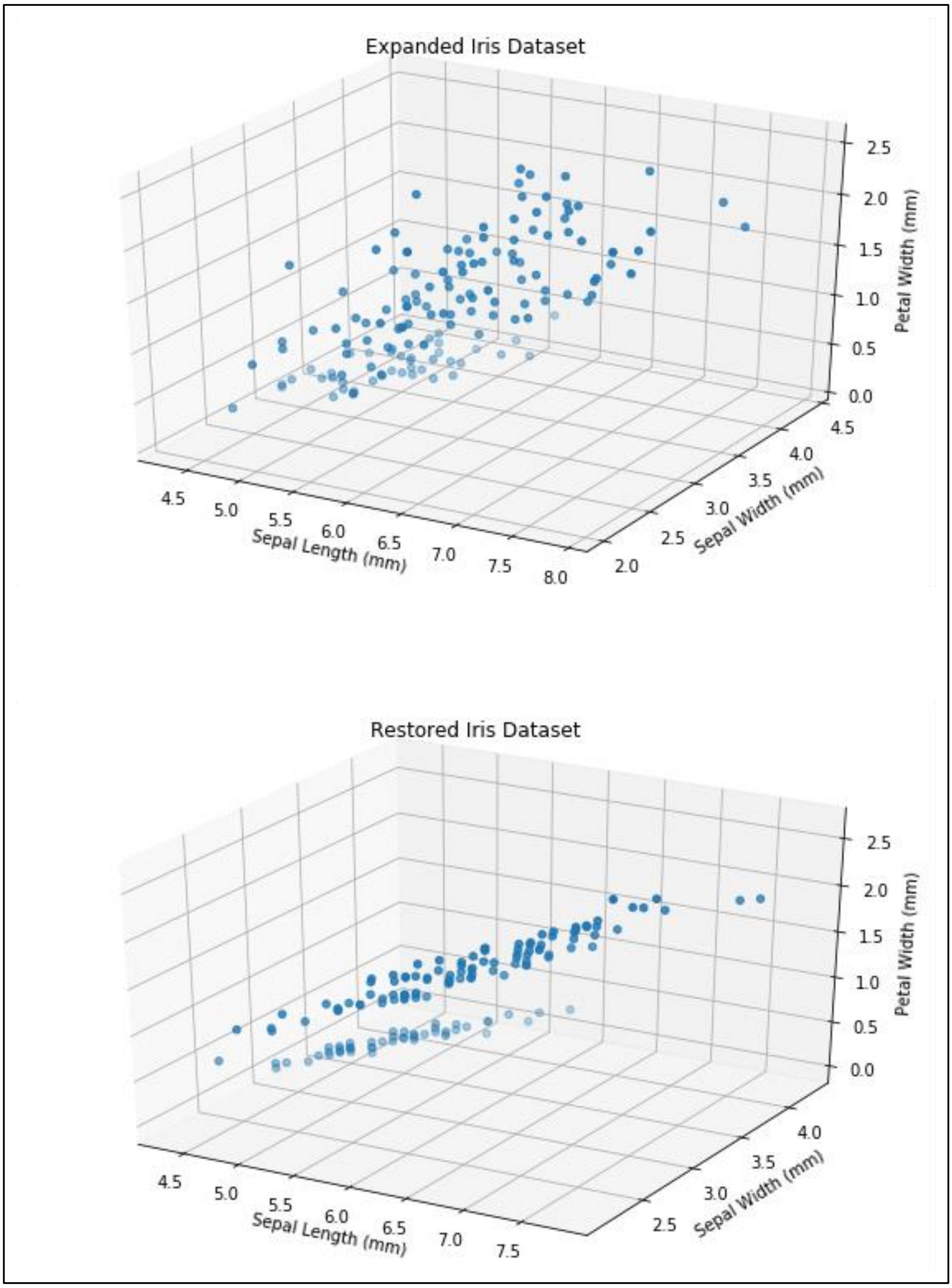


Figure 4.42: Expected plots

Figure 5.4: Autoencoder de-noising

Figure 5.5: Encoder/decoder representation

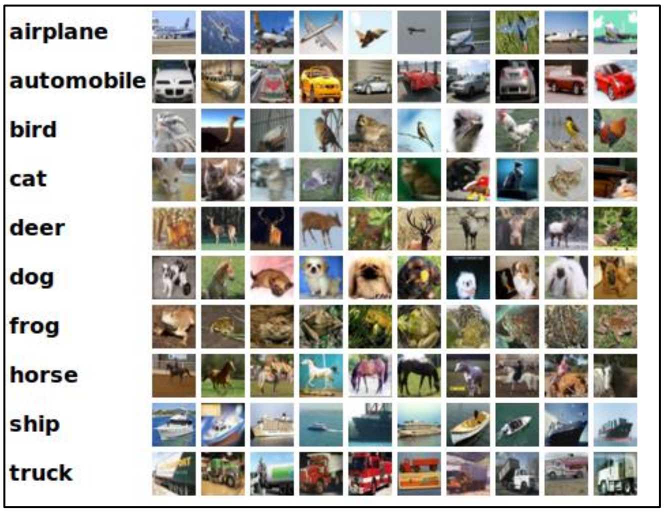


Figure 5.6: CIFAR-10 dataset

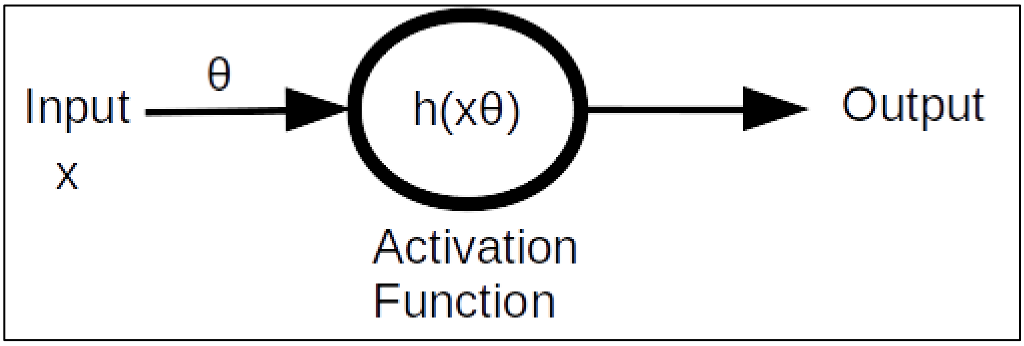


Figure 5.7: Anatomy of a neuron

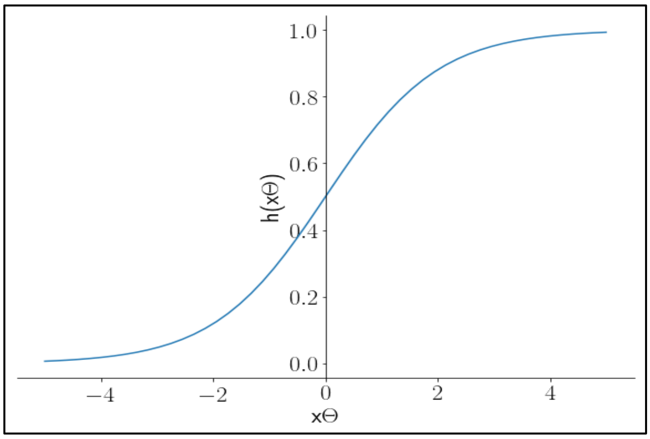


Figure 5.8: Output of the sigmoid function

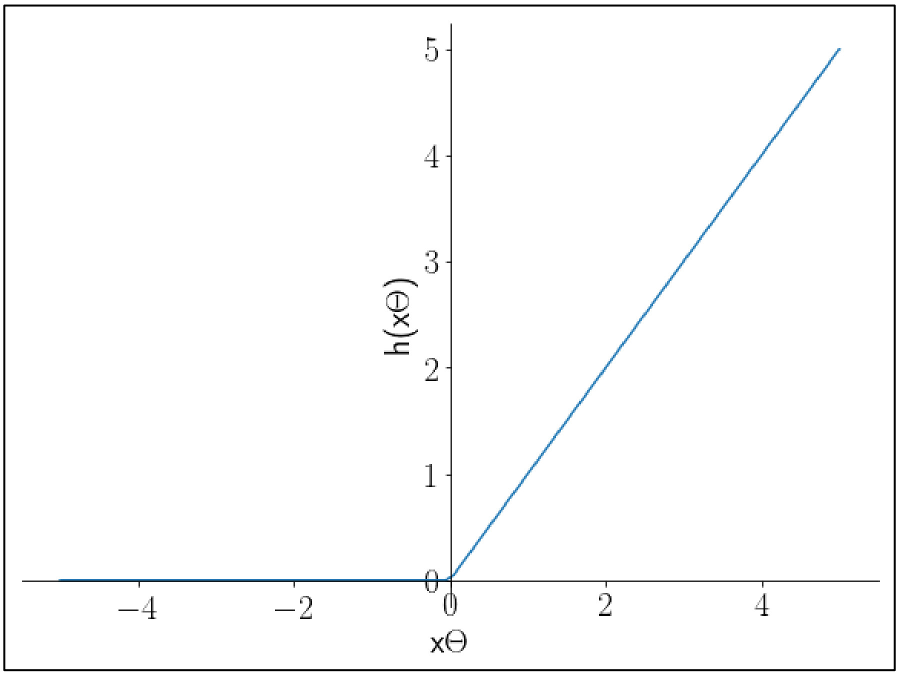


Figure 5.9: Output of ReLU

```
array([-5.        , -4.8989899 , -4.7979798 , -4.6969697 , -4.5959596 ,
       -4.49494949, -4.39393939, -4.29292929, -4.19191919, -4.09090909,
       -3.98989899, -3.88888889, -3.78787879, -3.68686869, -3.58585859,
       -3.48484848, -3.38383838, -3.28282828, -3.18181818, -3.08080808,
       -2.97979798, -2.87878788, -2.77777778, -2.67676768, -2.57575758,
       -2.47474747, -2.37373737, -2.27272727, -2.17171717, -2.07070707,
       -1.96969697, -1.86868687, -1.76767677, -1.66666667, -1.56565657,
```

Figure 5.7: Printing the inputs

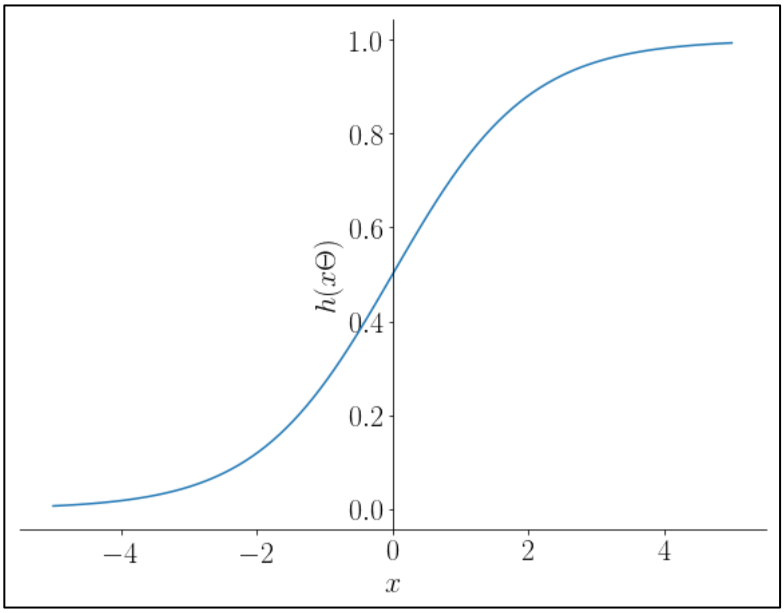


Figure 5.8: Plot of neurons versus inputs

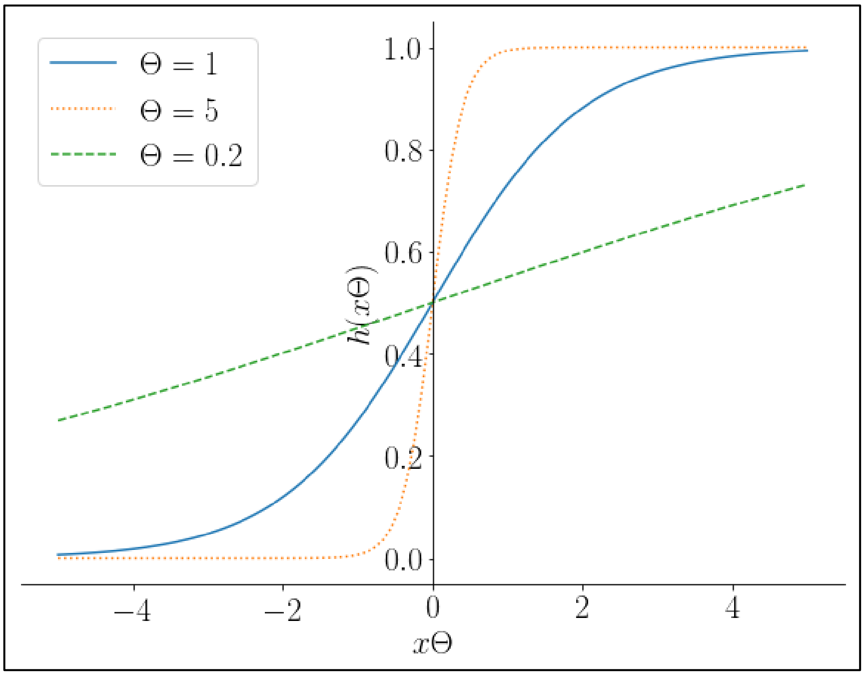


Figure 5.9: Output curves of neurons

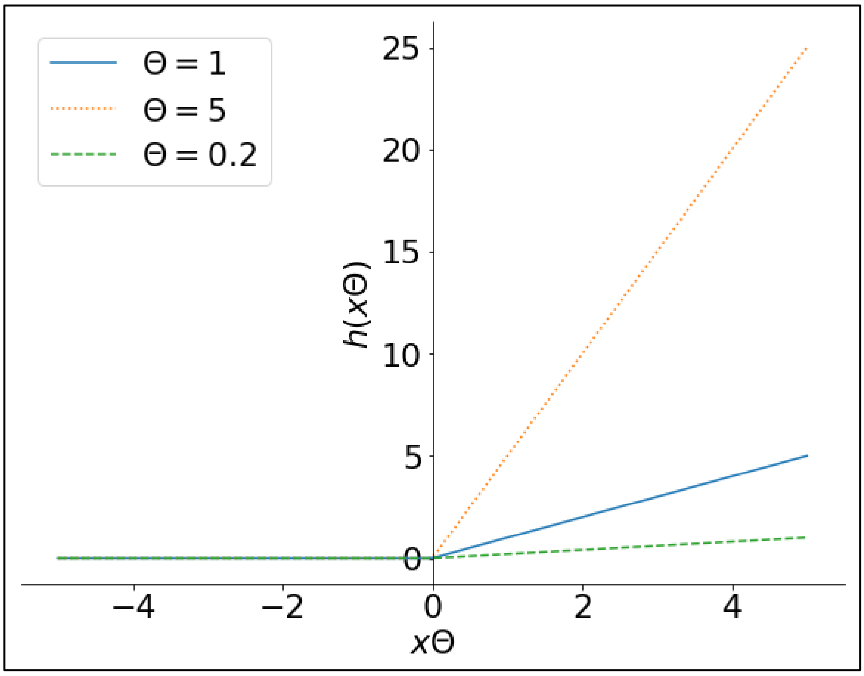


Figure 5.10: Expected output curves

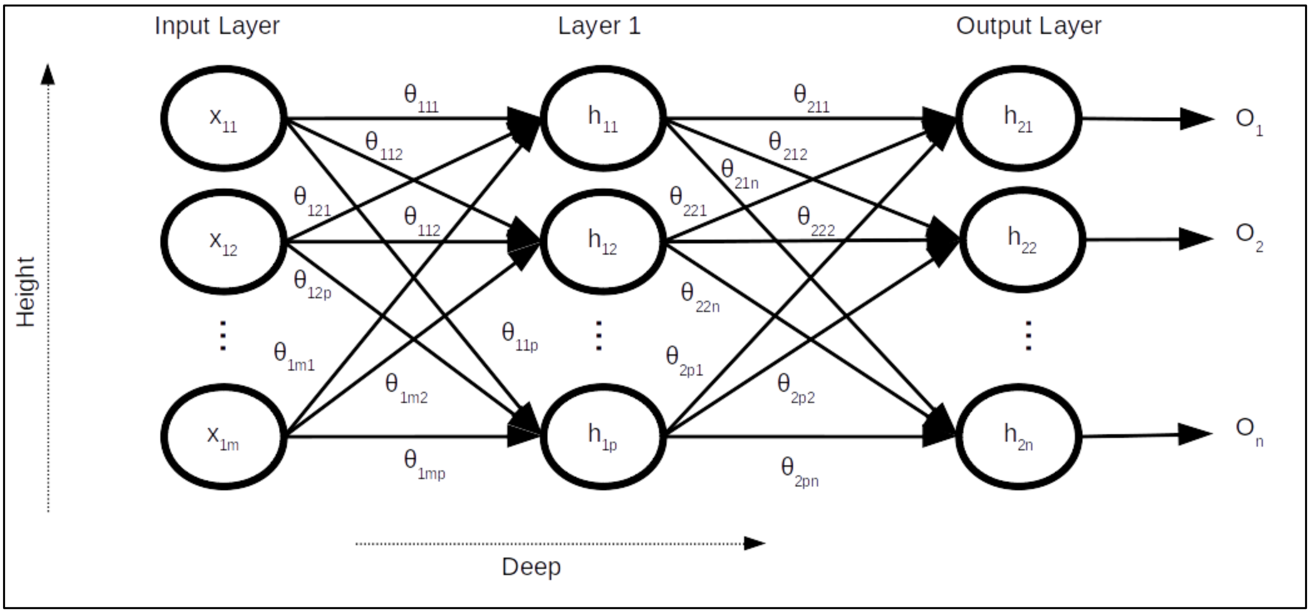


Figure 5.11: Simplified representation of a neural network

$$h_{11}(x_{11}\theta_{111} + x_{12}\theta_{121} + \ldots + x_{1m}\theta_{1m1})$$

Figure 5.12: Calculating the output of the last node

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 500)               512500
_________________________________________________________________
dense_2 (Dense)              (None, 10)                5010
=================================================================
Total params: 517,510
Trainable params: 517,510
Non-trainable params: 0
_________________________________________________________________
```

Figure 5.13: Structure and count of trainable parameters in the model

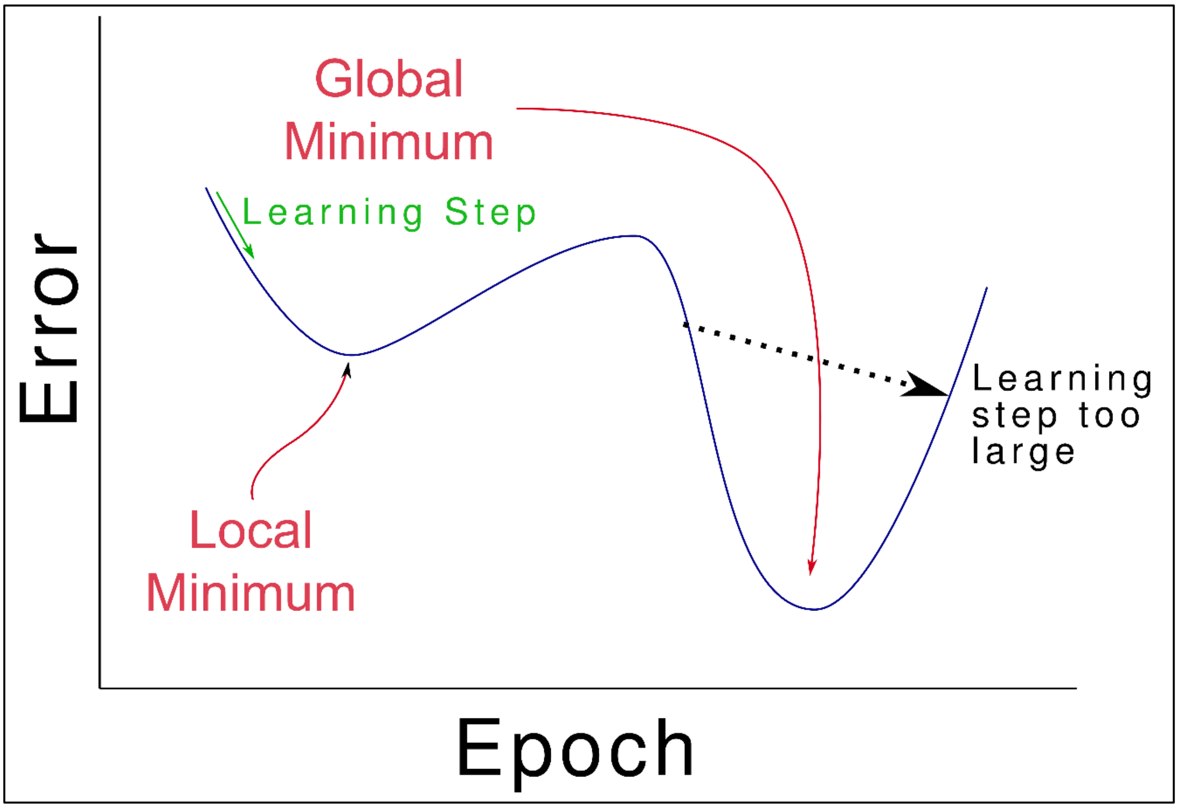


Figure 5.14: Selecting the correct learning rate (one epoch is one learning step)

```
[6,
 9,
 9,
 4,
 1,
 1,
 2,
 7,
 8,
 3,
 4,
 7,
```

Figure 5.15: Displaying the labels

```
array([[ 59,  43,  50, ..., 140,  84,  72],
       [154, 126, 105, ..., 139, 142, 144],
       [255, 253, 253, ...,  83,  83,  84],
       ...,
       [ 71,  60,  74, ...,  68,  69,  68],
       [250, 254, 211, ..., 215, 255, 254],
       [ 62,  61,  60, ..., 130, 130, 131]], dtype=uint8)
```

Figure 5.16: Content of the data key

Figure 5.17: The first 12 images

```
{b'num_cases_per_batch': 10000,
 b'label_names': [b'airplane',
  b'automobile',
  b'bird',
  b'cat',
  b'deer',
  b'dog',
  b'frog',
  b'horse',
  b'ship',
  b'truck'],
 b'num_vis': 3072}
```

Figure 5.18: Meaning of the labels

```
['airplane',
 'automobile',
 'bird',
 'cat',
 'deer',
 'dog',
 'frog',
 'horse',
 'ship',
 'truck']
```

Figure 5.19: Printing the actual labels

```
frog, truck, truck, deer, automobile, automobile, bird, horse, ship, cat,
deer, horse,
```

Figure 5.20: Labels of the first 12 images

```
array([[0., 0., 0., 0., 0., 0., 1., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 1.],
       [0., 0., 0., 0., 0., 0., 0., 0., 0., 1.],
       [0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 1., 0., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 1., 0., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 1., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 0., 1., 0.],
       [0., 0., 0., 1., 0., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 1., 0., 0., 0., 0., 0.],
       [0., 0., 0., 0., 0., 0., 0., 1., 0., 0.]])
```

Figure 5.21: One hot encoding values for first 12 samples

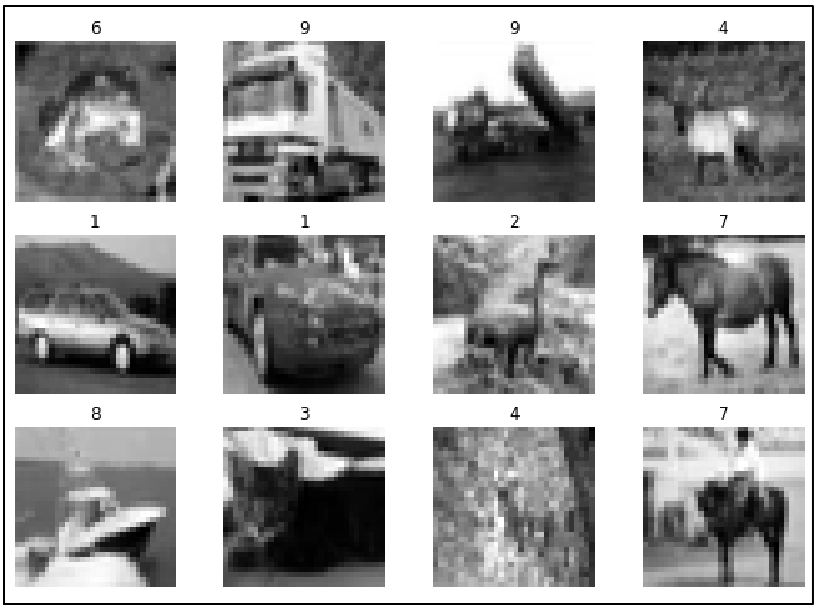


Figure 5.22: Displaying the first 12 images again.

```
Epoch 97/100
10000/10000 [==============================] - 2s 178us/step - loss: 0.4526 - acc: 0.8824
Epoch 98/100
10000/10000 [==============================] - 2s 176us/step - loss: 0.4488 - acc: 0.8871
Epoch 99/100
10000/10000 [==============================] - 2s 174us/step - loss: 0.4384 - acc: 0.8940
Epoch 100/100
10000/10000 [==============================] - 2s 170us/step - loss: 0.4322 - acc: 0.8955

<keras.callbacks.History at 0x7f12b3c7b978>
```

Figure 5.23: Training the model

```
array([[2.72101886e-03, 2.82521220e-03, 5.80681080e-04, 2.00835592e-03,
        4.87272721e-03, 1.73771027e-02, 9.62930799e-01, 5.69747109e-03,
        7.23911216e-04, 2.62686226e-04],
       [3.00214946e-04, 1.14106536e-01, 4.17048521e-02, 1.38805415e-02,
        4.82545962e-04, 3.11067980e-02, 1.02459533e-04, 2.45292974e-03,
        6.33396162e-03, 7.89529204e-01],
       [1.38785226e-05, 7.90050399e-05, 4.03187078e-05, 1.66309916e-03,
        3.49369337e-04, 3.01616683e-06, 5.77264291e-06, 3.29075777e-03,
        2.98287741e-05, 9.94524956e-01],
```

Figure 5.24: Printing the predictions

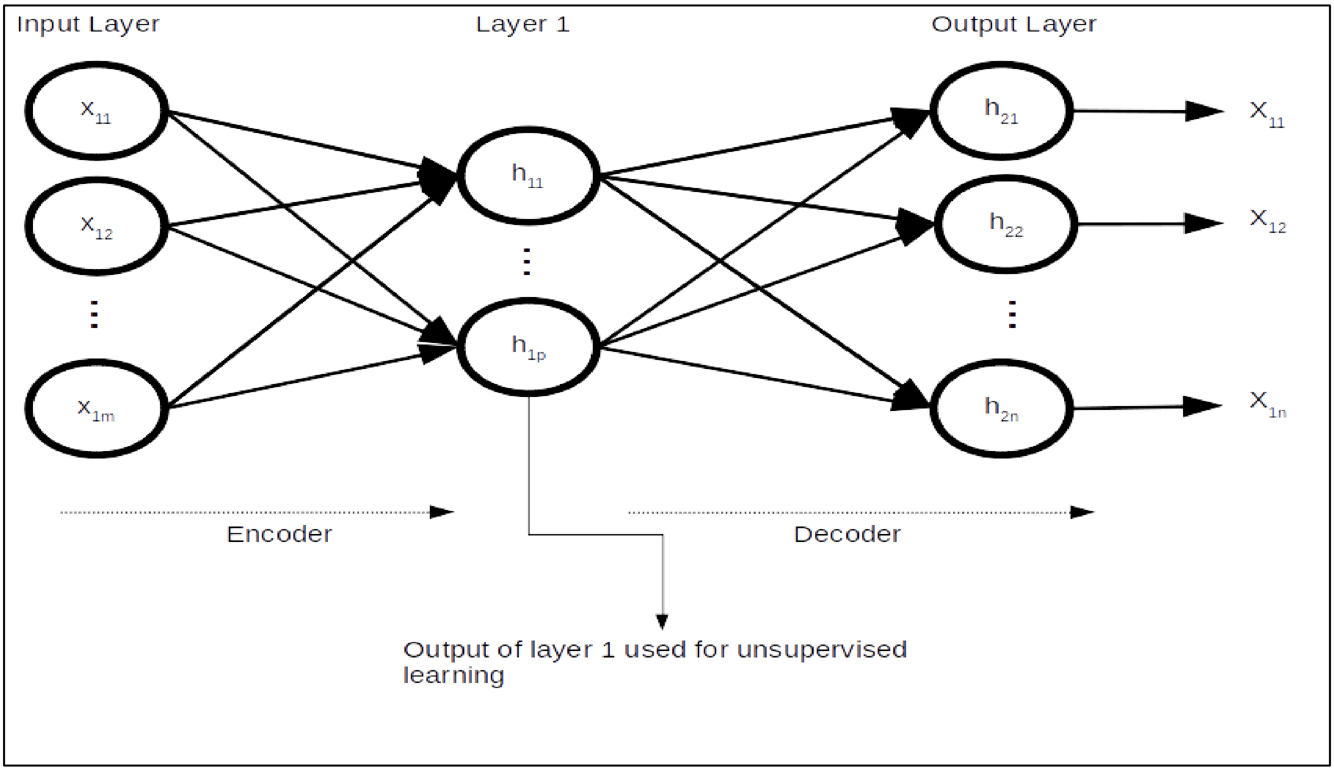


Figure 5.25: Simple autoencoder network architecture

```
Epoch 95/100
10000/10000 [==============================] - 4s 416us/step - loss: 0.5779
Epoch 96/100
10000/10000 [==============================] - 4s 418us/step - loss: 0.5777
Epoch 97/100
10000/10000 [==============================] - 4s 434us/step - loss: 0.5778
Epoch 98/100
10000/10000 [==============================] - 4s 428us/step - loss: 0.5776
Epoch 99/100
10000/10000 [==============================] - 4s 438us/step - loss: 0.5775
Epoch 100/100
10000/10000 [==============================] - 4s 404us/step - loss: 0.5775

<keras.callbacks.History at 0x7fb44d6fe8d0>
```

Figure 5.26: Training the model

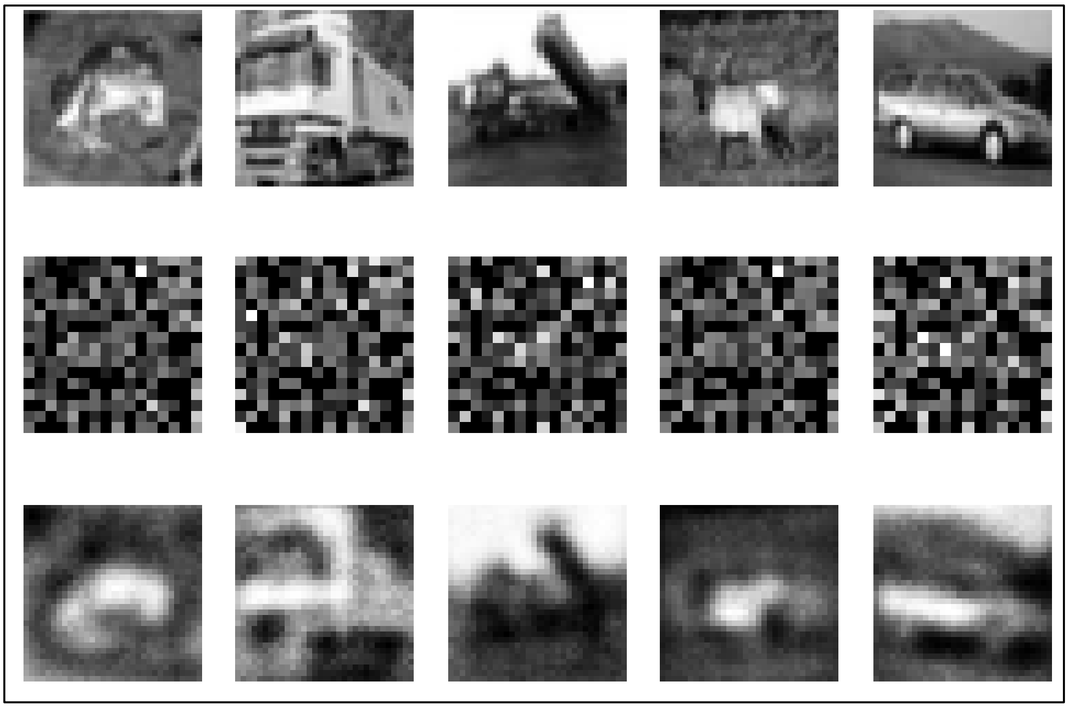


Figure 5.27: Output of simple autoencoder

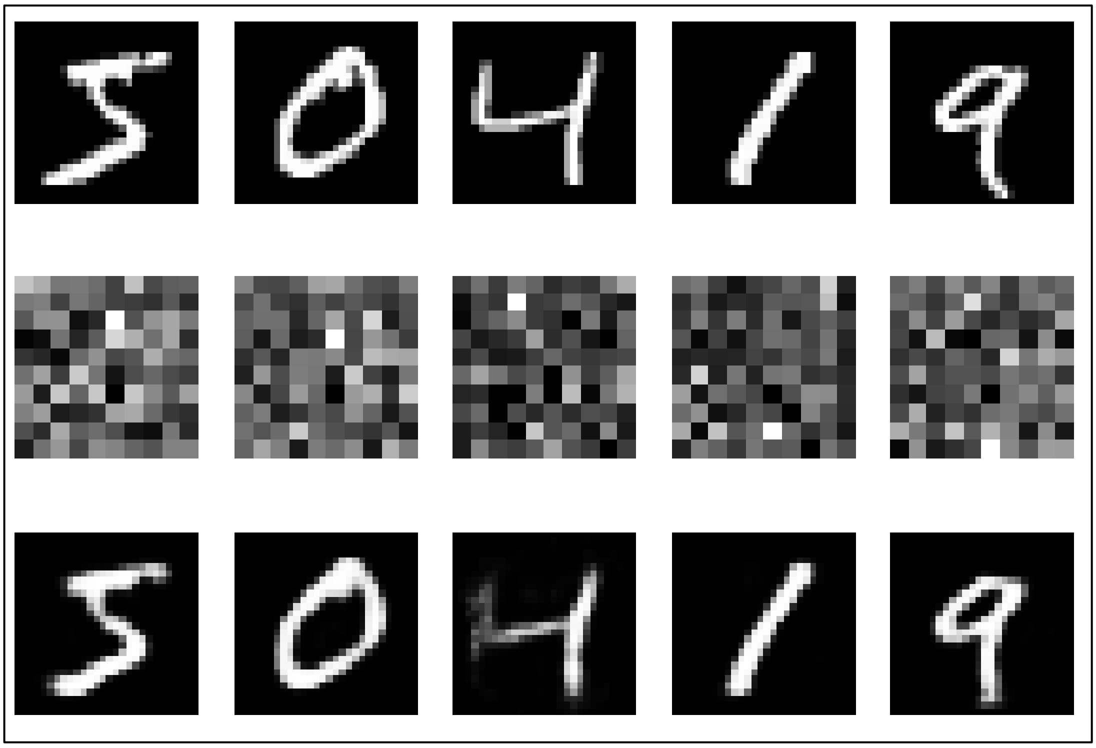


Figure 5.28: Expected plot of original image, the encoder output, and the decoder

```
Epoch 93/100
10000/10000 [==============================] - 9s 945us/step - loss: 0.5805
Epoch 94/100
10000/10000 [==============================] - 10s 965us/step - loss: 0.5806
Epoch 95/100
10000/10000 [==============================] - 10s 969us/step - loss: 0.5807
Epoch 96/100
10000/10000 [==============================] - 10s 968us/step - loss: 0.5804
Epoch 97/100
10000/10000 [==============================] - 10s 1ms/step - loss: 0.5803
Epoch 98/100
10000/10000 [==============================] - 10s 971us/step - loss: 0.5804
Epoch 99/100
10000/10000 [==============================] - 10s 970us/step - loss: 0.5802
Epoch 100/100
10000/10000 [==============================] - 10s 972us/step - loss: 0.5799
```

Figure 5.29: Training the model

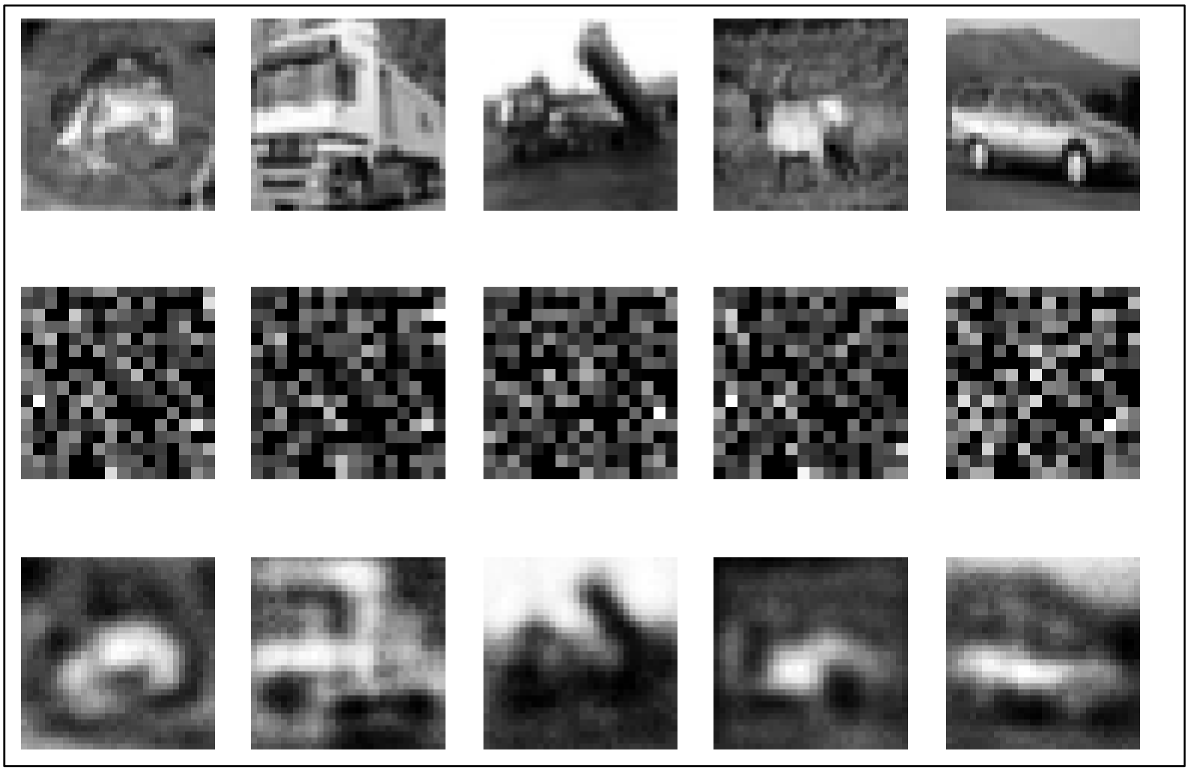


Figure 5.30: Output of multi-layer autoencoder

1	1
1	2

Figure 5.31: Demonstration of sample matrix

```
Epoch 1/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5934
Epoch 2/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5687
Epoch 3/20
10000/10000 [==============================] - 22s 2ms/step - loss: 0.5633
Epoch 4/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5602
Epoch 5/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5590:
Epoch 6/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5581
Epoch 7/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5578
Epoch 8/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5572
Epoch 9/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5566
Epoch 10/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5557
Epoch 11/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5553
Epoch 12/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5552
Epoch 13/20
10000/10000 [==============================] - 21s 2ms/step - loss: 0.5551
Epoch 14/20
10000/10000 [==============================] - 22s 2ms/step - loss: 0.5543
Epoch 15/20
10000/10000 [==============================] - 22s 2ms/step - loss: 0.5544
Epoch 16/20
10000/10000 [==============================] - 22s 2ms/step - loss: 0.5548
Epoch 17/20
10000/10000 [==============================] - 22s 2ms/step - loss: 0.5541
Epoch 18/20
10000/10000 [==============================] - 22s 2ms/step - loss: 0.5539
Epoch 19/20
10000/10000 [==============================] - 22s 2ms/step - loss: 0.5538
Epoch 20/20
10000/10000 [==============================] - 22s 2ms/step - loss: 0.5539
```

Figure 5.32: Training the model

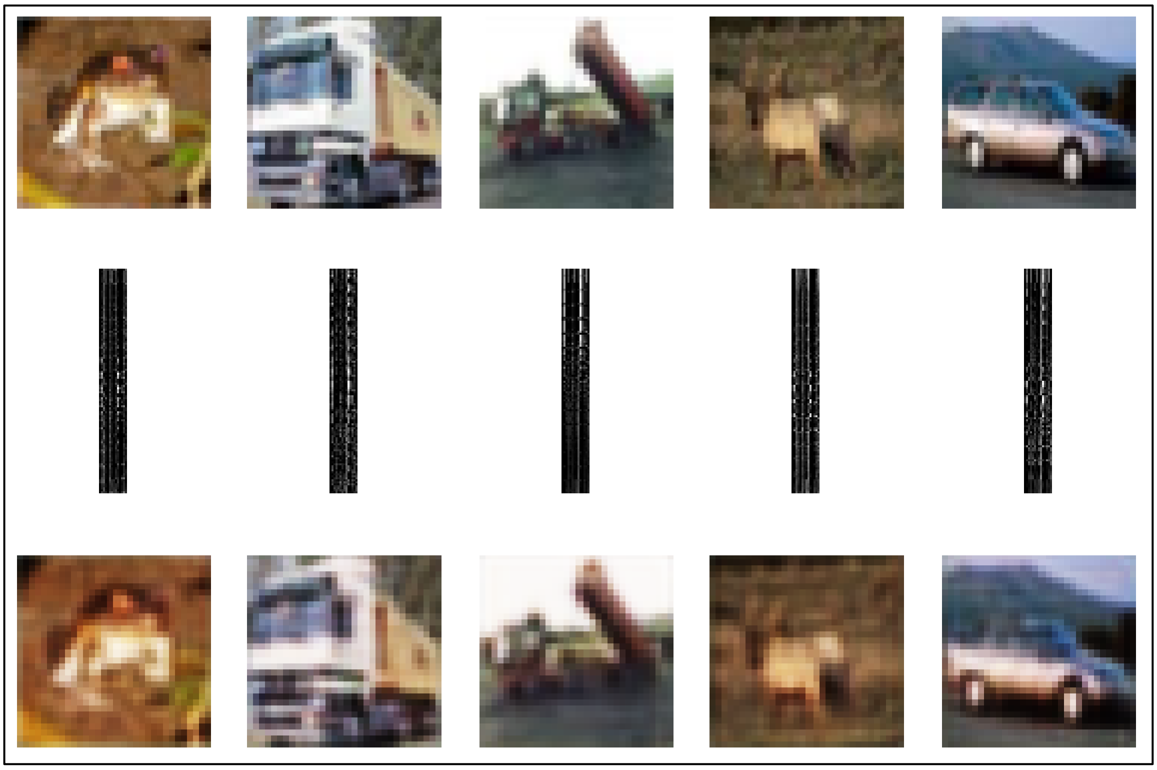


Figure 5.33: The original image, the encoder output, and the decoder

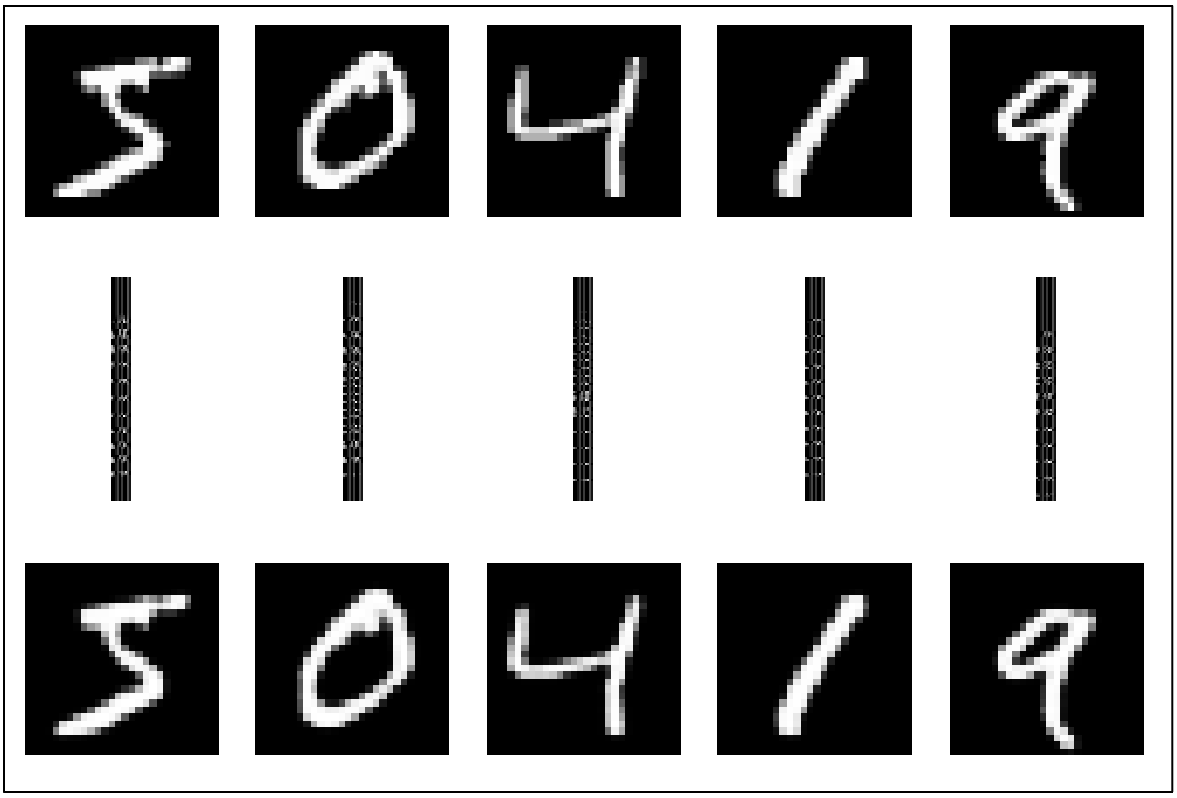


Figure 5.34: Expected original image, the encoder output, and the decoder

Lesson 6: t-Distributed Stochastic Neighbor Embedding (t-SNE)

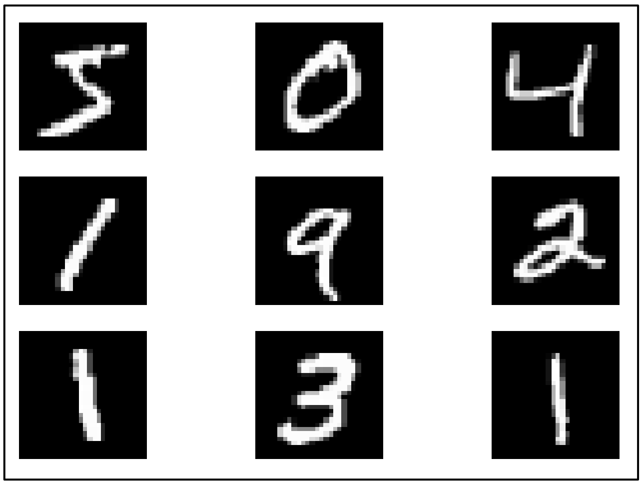


Figure 6.10: MNIST data sample

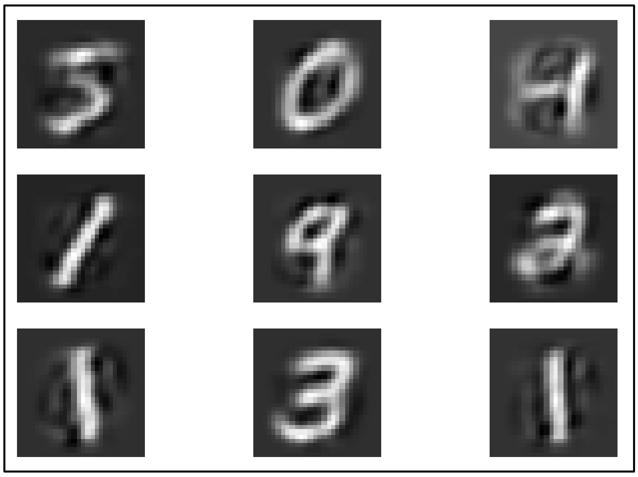


Figure 6.11: MNST reduced using PCA to 30 components

$$C = \sum_i \sum_j p_{i|j} \log \frac{p_{i|j}}{q_{i|j}}$$

Figure 6.12: Kullback-Leibler divergence.

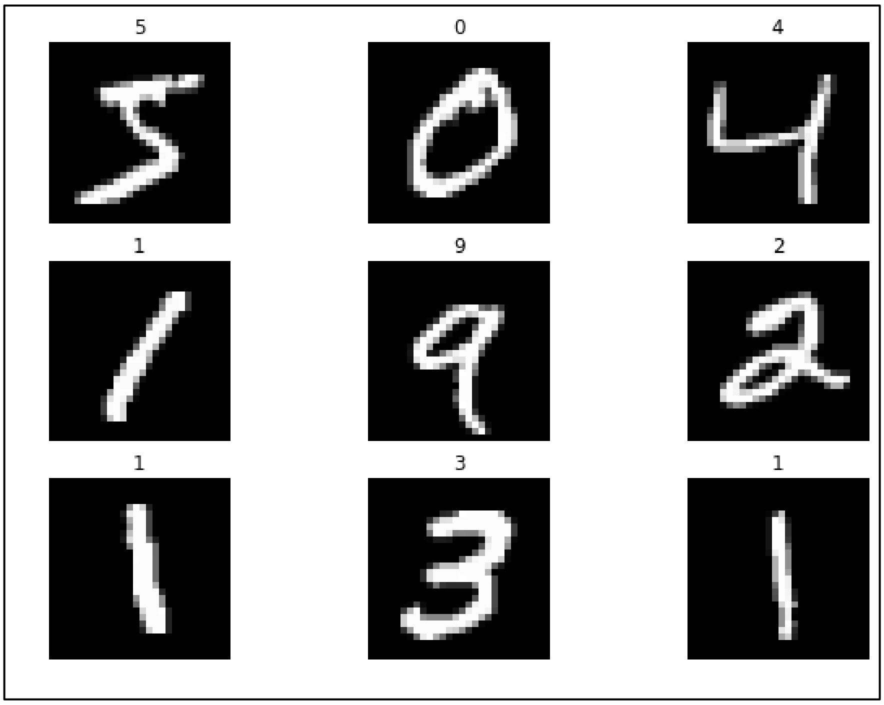


Figure 6.4: Output after loading the dataset

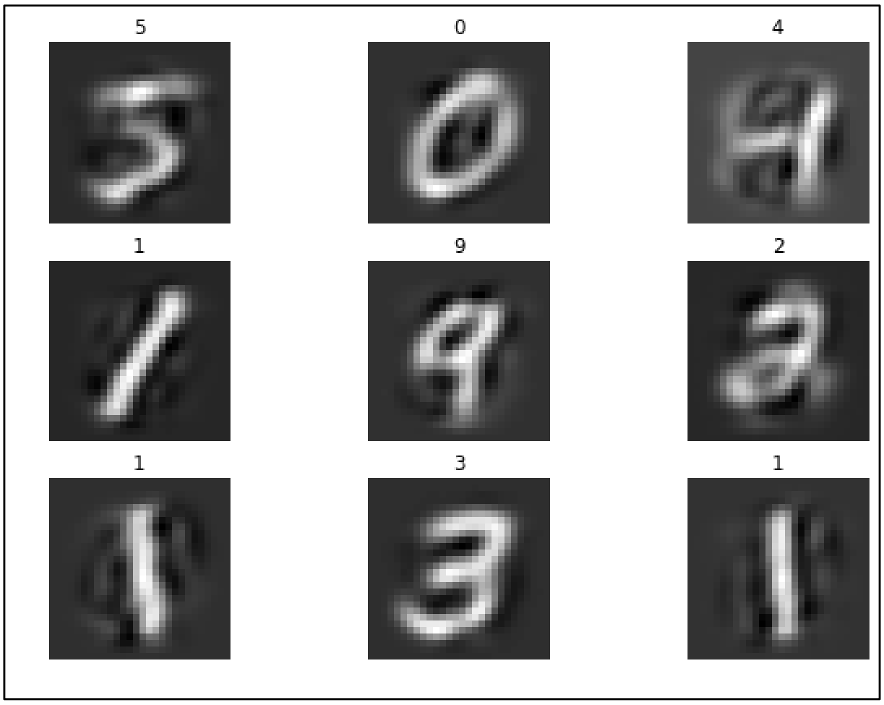


Figure 6.5: Visualizing the effect of reducing the dataset

```
TSNE(angle=0.5, early_exaggeration=12.0, init='random', learning_rate=200.0,
   method='barnes_hut', metric='euclidean', min_grad_norm=1e-07,
   n_components=2, n_iter=1000, n_iter_without_progress=300,
   perplexity=30.0, random_state=None, verbose=0)
```

Figure 6.6: Applying t-SNE to PCA-transformed data

```
[t-SNE] Computing 91 nearest neighbors...
[t-SNE] Indexed 10000 samples in 0.016s...
[t-SNE] Computed neighbors for 10000 samples in 5.454s...
[t-SNE] Computed conditional probabilities for sample 1000 / 10000
[t-SNE] Computed conditional probabilities for sample 2000 / 10000
[t-SNE] Computed conditional probabilities for sample 3000 / 10000
[t-SNE] Computed conditional probabilities for sample 4000 / 10000
[t-SNE] Computed conditional probabilities for sample 5000 / 10000
[t-SNE] Computed conditional probabilities for sample 6000 / 10000
[t-SNE] Computed conditional probabilities for sample 7000 / 10000
[t-SNE] Computed conditional probabilities for sample 8000 / 10000
[t-SNE] Computed conditional probabilities for sample 9000 / 10000
[t-SNE] Computed conditional probabilities for sample 10000 / 10000
[t-SNE] Mean sigma: 304.998835
[t-SNE] KL divergence after 250 iterations with early exaggeration: 85.546951
[t-SNE] KL divergence after 1000 iterations: 1.696535
```

Figure 6.7: Transforming the decomposed dataset

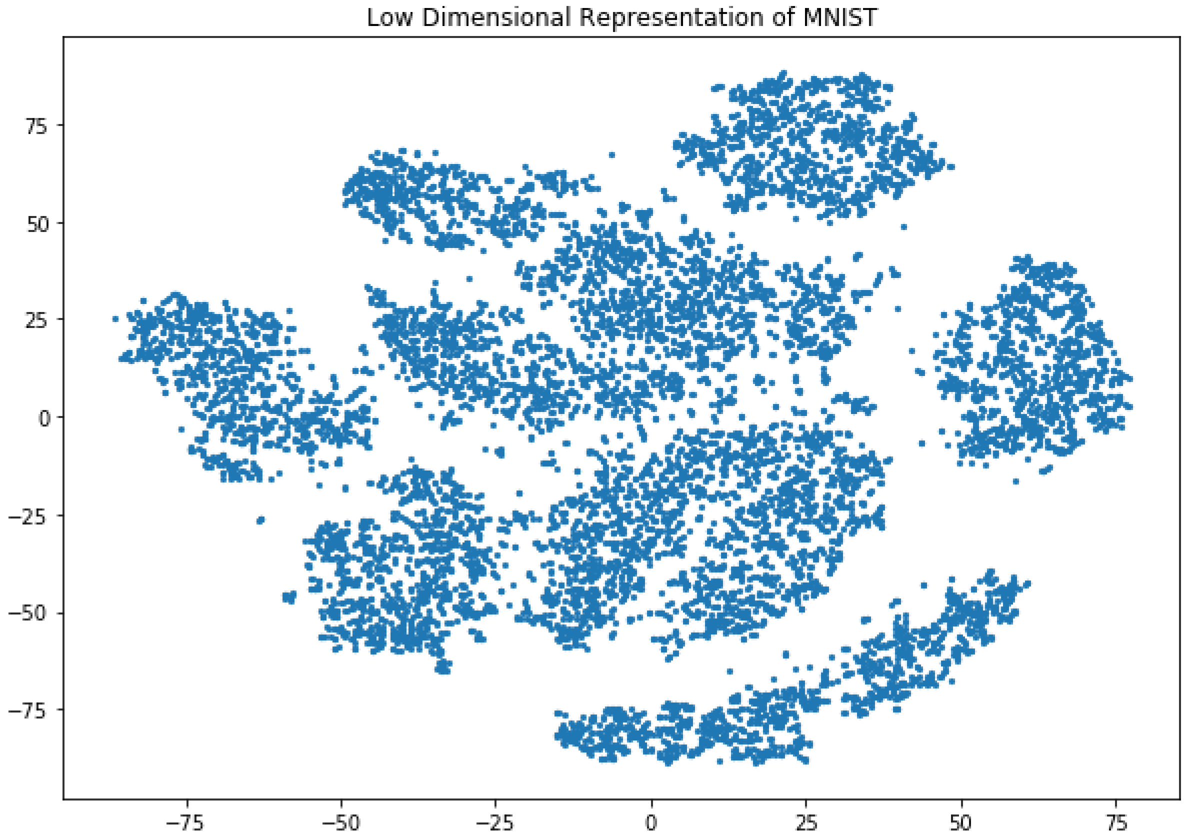


Figure 13.8: 2D representation of MNIST (no labels).

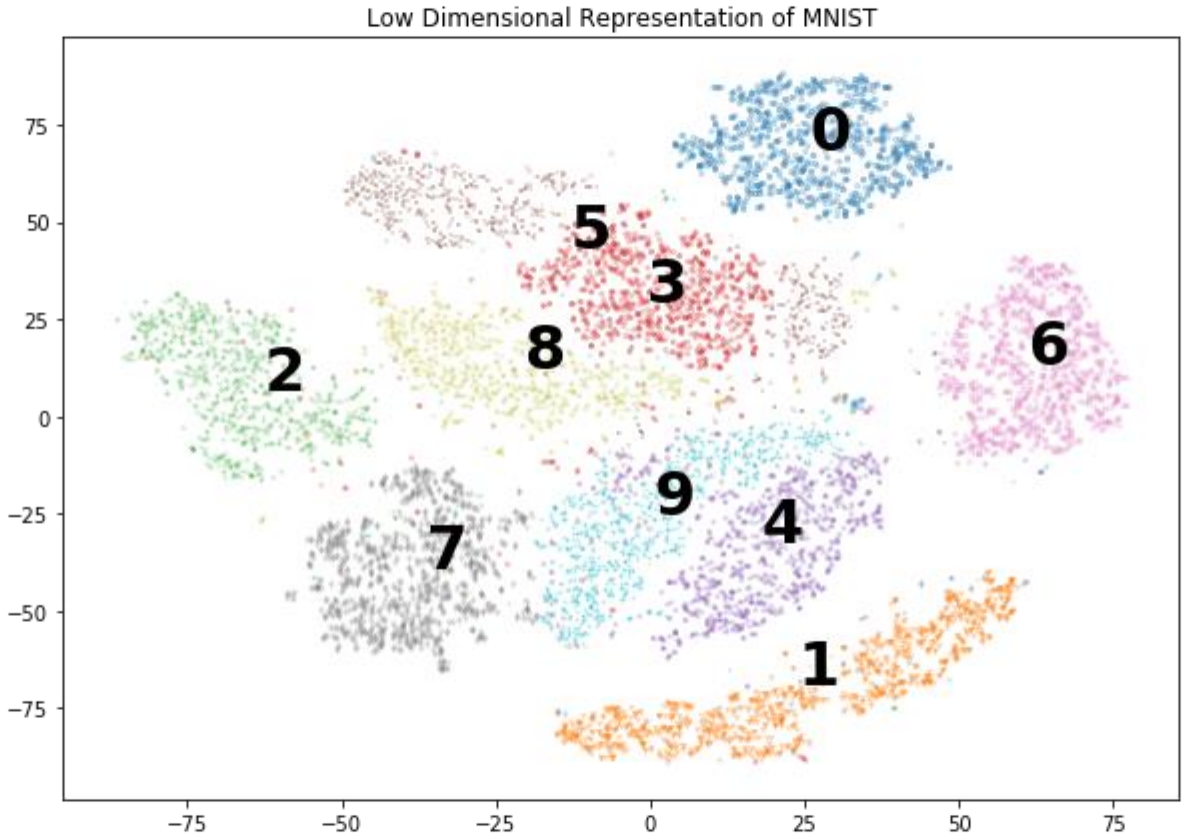


Figure 6.9: 2D representation of MNIST with labels.

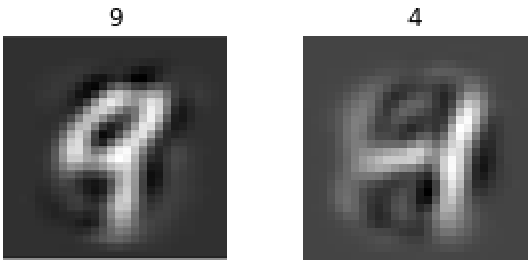


Figure 6.10: PCA images of nine.

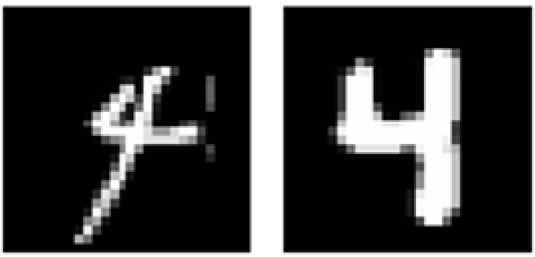


Figure 6.11: Shape of number four

```
array([   7,   10,   12, ..., 9974, 9977, 9991])
```

Figure 6.12: Index of threes in the dataset.

```
array([   0,    1,    6,   11,   13,   14,   17,   18,   19,   21,   22,
         23,   25,   29,   30,   31,   32,   34,   35,   37,   38,   39,
         41,   42,   43,   45,   50,   51,   52,   54,   55,   56,   57,
         58,   59,   60,   61,   62,   63,   66,   67,   68,   71,   72,
```

Figure 6.13: The threes with x value less than zero.

```
array([[-16.126516 ,  35.23472  ],
       [ -4.217844 ,  31.871649 ],
       [ -2.3769686,  35.472614 ],
       ...,
       [ -6.4078546,  38.2851   ],
       [-10.40415  ,  45.599823 ],
       [ -8.813534 ,  39.997196 ]], dtype=float32)
```

Figure 6.14: Coordinates away from the three cluster

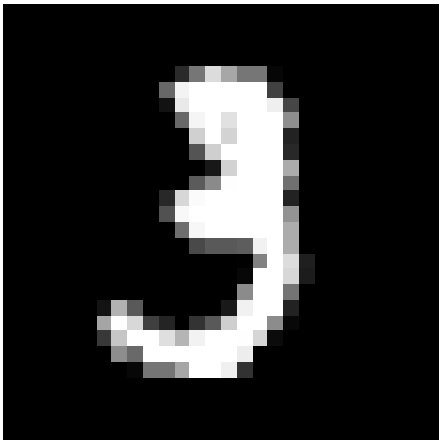


Figure 6.15: Image of sample ten

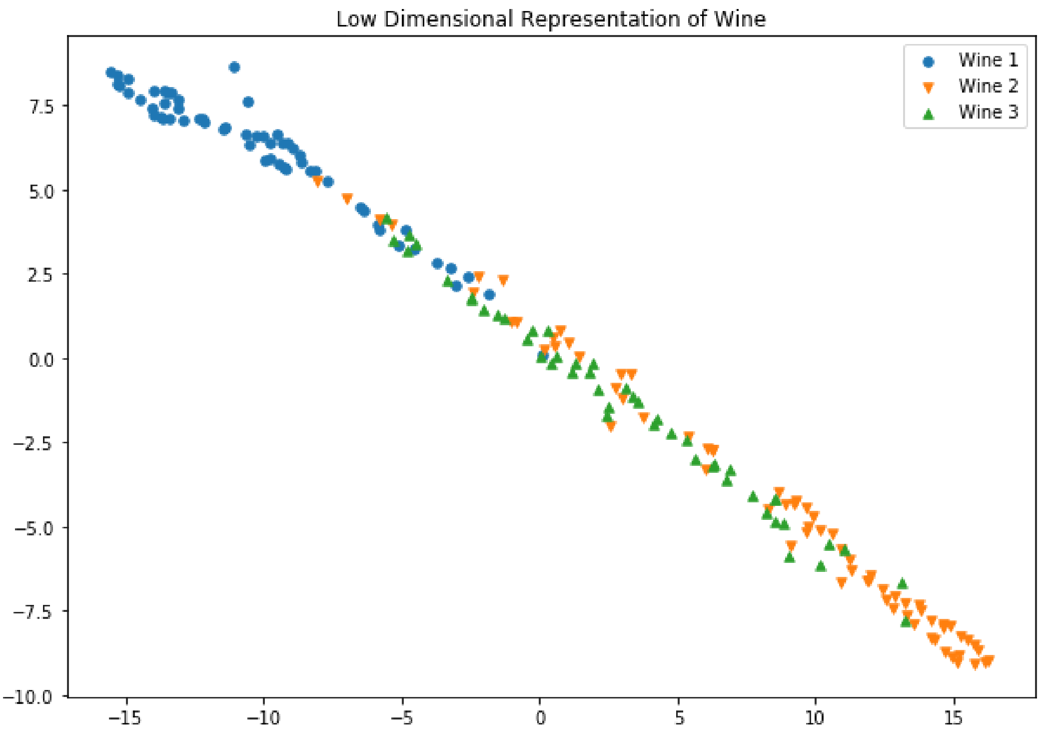


Figure 6.16: The expected plot

```
[t-SNE] Computing 10 nearest neighbors...
[t-SNE] Indexed 10000 samples in 0.018s...
[t-SNE] Computed neighbors for 10000 samples in 3.438s...
[t-SNE] Computed conditional probabilities for sample 1000 / 10000
[t-SNE] Computed conditional probabilities for sample 2000 / 10000
[t-SNE] Computed conditional probabilities for sample 3000 / 10000
[t-SNE] Computed conditional probabilities for sample 4000 / 10000
[t-SNE] Computed conditional probabilities for sample 5000 / 10000
[t-SNE] Computed conditional probabilities for sample 6000 / 10000
[t-SNE] Computed conditional probabilities for sample 7000 / 10000
[t-SNE] Computed conditional probabilities for sample 8000 / 10000
[t-SNE] Computed conditional probabilities for sample 9000 / 10000
[t-SNE] Computed conditional probabilities for sample 10000 / 10000
[t-SNE] Mean sigma: 165.134196
[t-SNE] KL divergence after 250 iterations with early exaggeration: 96.804878
[t-SNE] KL divergence after 1000 iterations: 1.850921
[t-SNE] Computing 91 nearest neighbors...
[t-SNE] Indexed 10000 samples in 0.014s...
[t-SNE] Computed neighbors for 10000 samples in 5.129s...
[t-SNE] Computed conditional probabilities for sample 1000 / 10000
[t-SNE] Computed conditional probabilities for sample 2000 / 10000
[t-SNE] Computed conditional probabilities for sample 3000 / 10000
[t-SNE] Computed conditional probabilities for sample 4000 / 10000
[t-SNE] Computed conditional probabilities for sample 5000 / 10000
[t-SNE] Computed conditional probabilities for sample 6000 / 10000
[t-SNE] Computed conditional probabilities for sample 7000 / 10000
[t-SNE] Computed conditional probabilities for sample 8000 / 10000
[t-SNE] Computed conditional probabilities for sample 9000 / 10000
[t-SNE] Computed conditional probabilities for sample 10000 / 10000
[t-SNE] Mean sigma: 283.586365
[t-SNE] KL divergence after 250 iterations with early exaggeration: 85.399399
[t-SNE] KL divergence after 1000 iterations: 1.696069
[t-SNE] Computing 901 nearest neighbors...
[t-SNE] Indexed 10000 samples in 0.013s...
[t-SNE] Computed neighbors for 10000 samples in 7.993s...
[t-SNE] Computed conditional probabilities for sample 1000 / 10000
[t-SNE] Computed conditional probabilities for sample 2000 / 10000
[t-SNE] Computed conditional probabilities for sample 3000 / 10000
[t-SNE] Computed conditional probabilities for sample 4000 / 10000
[t-SNE] Computed conditional probabilities for sample 5000 / 10000
[t-SNE] Computed conditional probabilities for sample 6000 / 10000
[t-SNE] Computed conditional probabilities for sample 7000 / 10000
[t-SNE] Computed conditional probabilities for sample 8000 / 10000
[t-SNE] Computed conditional probabilities for sample 9000 / 10000
[t-SNE] Computed conditional probabilities for sample 10000 / 10000
[t-SNE] Mean sigma: 393.939776
[t-SNE] KL divergence after 250 iterations with early exaggeration: 67.932961
[t-SNE] KL divergence after 1000 iterations: 1.193975
```

Figure 6.17: Iterating through a model

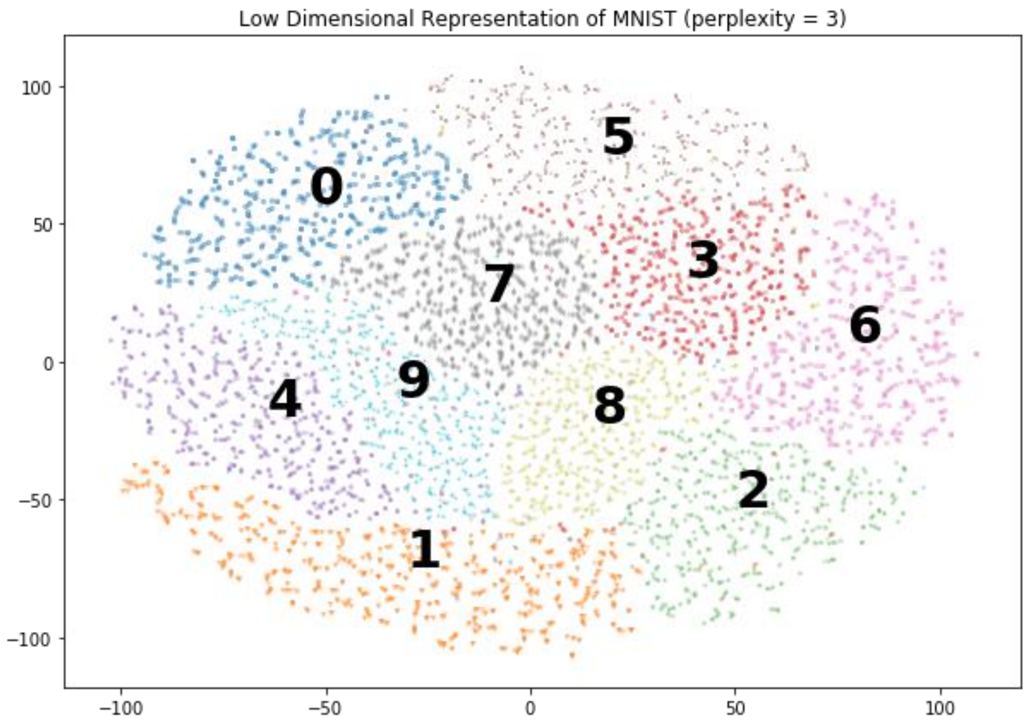


Figure 6.18: Plot of low perplexity value

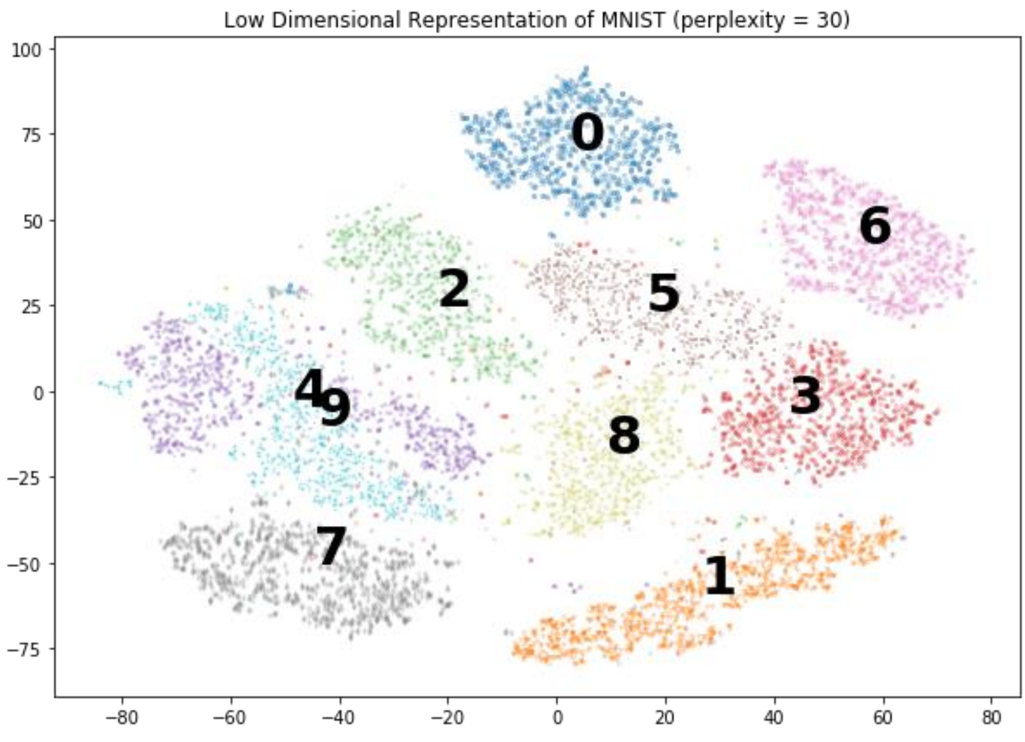


Figure 6.19: Plot after increasing perplexity by a factor of 10

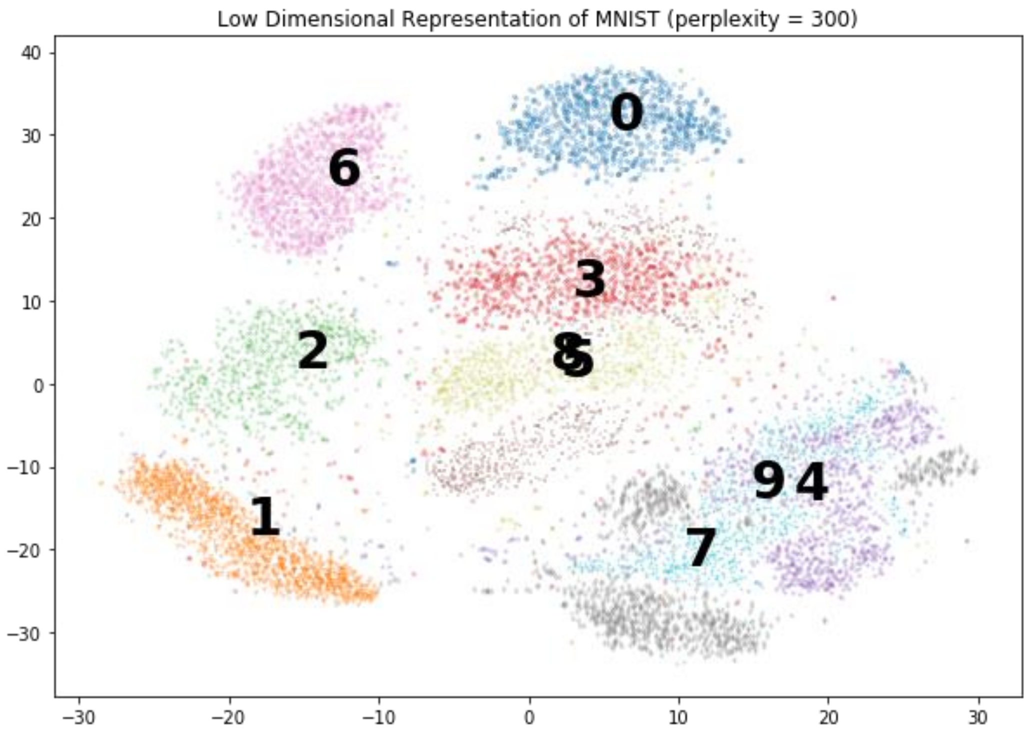


Figure 6.20: Increasing the perplexity value to 300

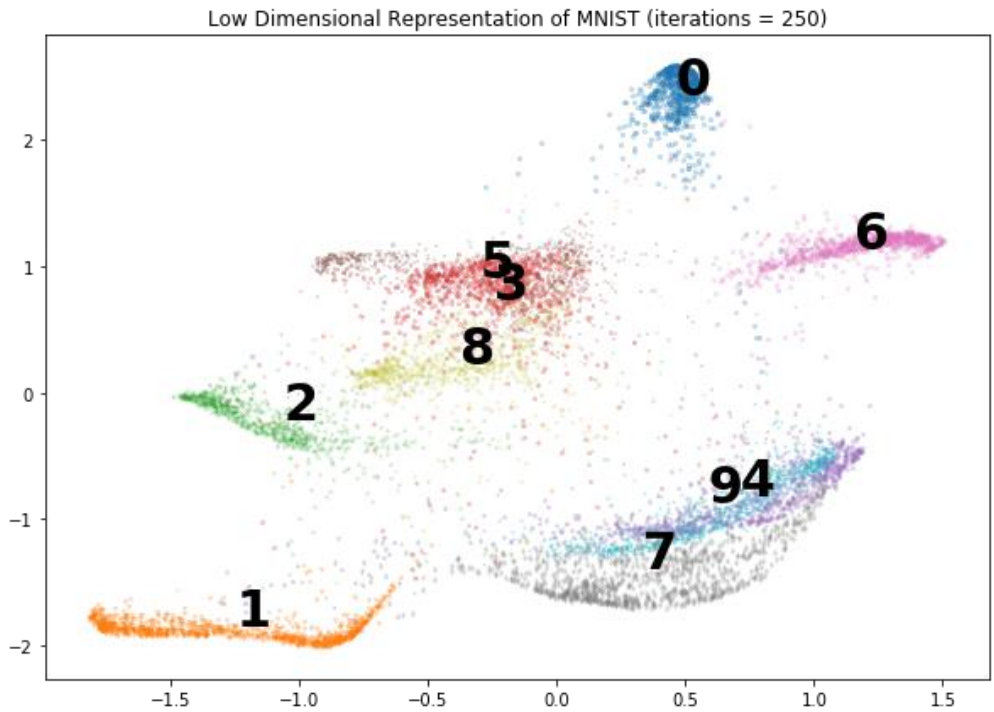


Figure 6.21: Plot after 250 iterations

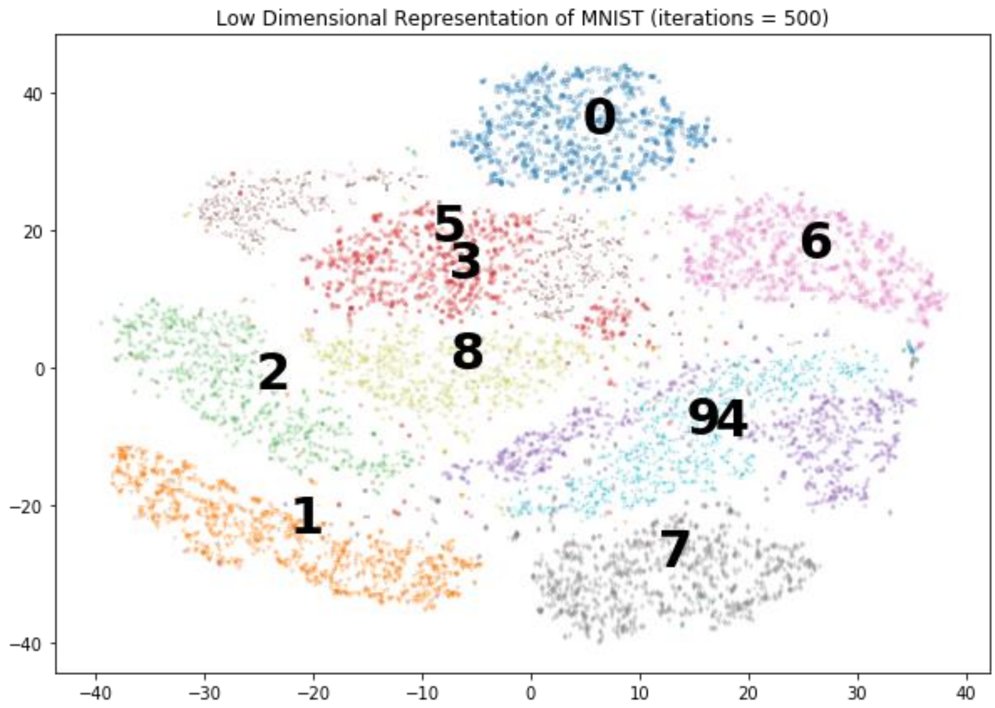


Figure 6.22: Plot after increasing the iterations to 500

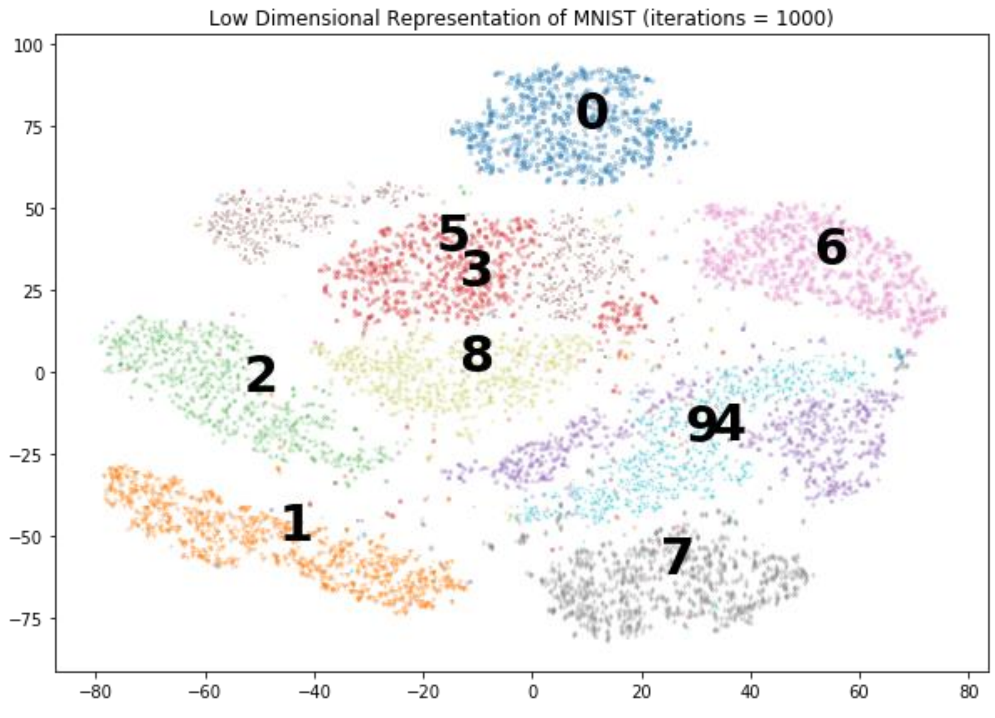


Figure 6.23: Plot after 1,000 iterations

Lesson 7: Topic Modeling

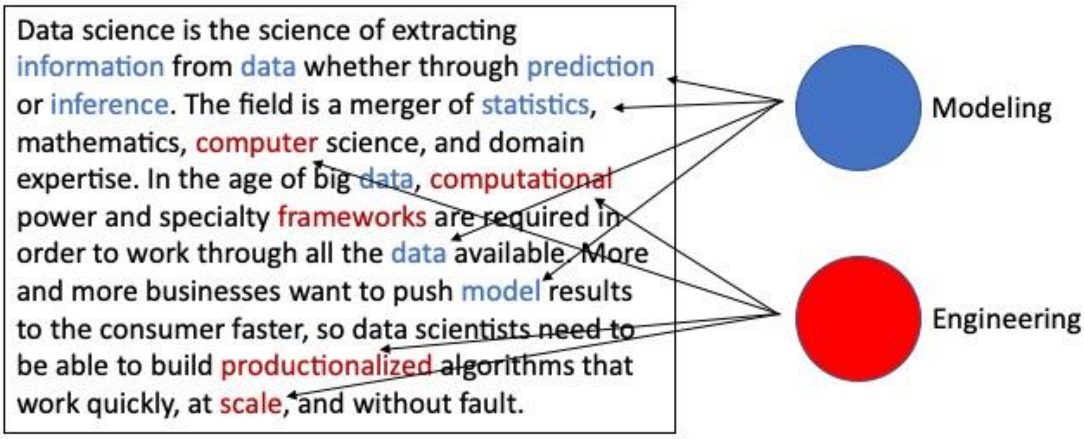


Figure 7.1: Example of identifying words in a text and assigning them to topics

Library	**Use**
`langdetect`	Used to detect the language of any text.
`matplotlib.pyplot`	Used to do basic plotting.
`nltk`	Used to do a variety of natural language processing tasks.
`numpy`	Used to work with arrays and matrices.
`pandas`	Used to work with data frames.
`pyLDAvis`	Used to visualize the results of Latent Dirichlet Allocation models.
`pyLDAvis.sklearn`	Used to run `pyLDAvis` with `sklearn` models.
`regex`	Used to write and execute regular expressions.
`sklearn`	Used to build machine learning models.

Figure 7.2: Table showing different libraries and their use

```
---------------------------------------------------------------------------
ModuleNotFoundError                          Traceback (most recent call last)
<ipython-input-3-a62286ae48f9> in <module>
      4 import numpy
      5 import pandas
----> 6 import pyLDAvis
      7 import pyLDAvis.sklearn
      8 import regex

ModuleNotFoundError: No module named 'pyLDAvis'
```

Figure 7.3: Library not installed error

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\rutujay\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\wordnet.zip.
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\rutujay\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.

True
```

Figure 7.4: Importing libraries and downloading dictionaries

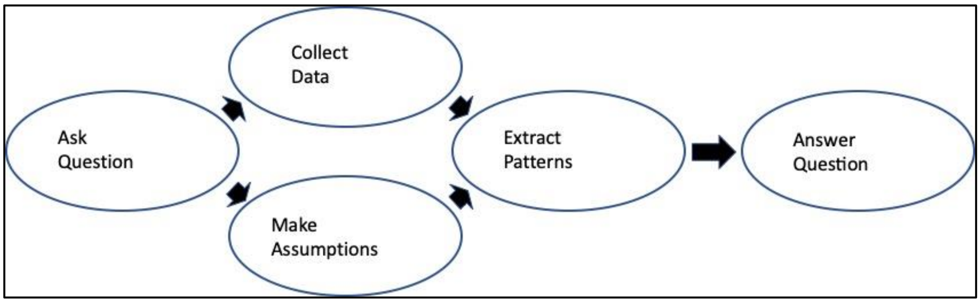


Figure 7.5: The generic topic modeling workflow

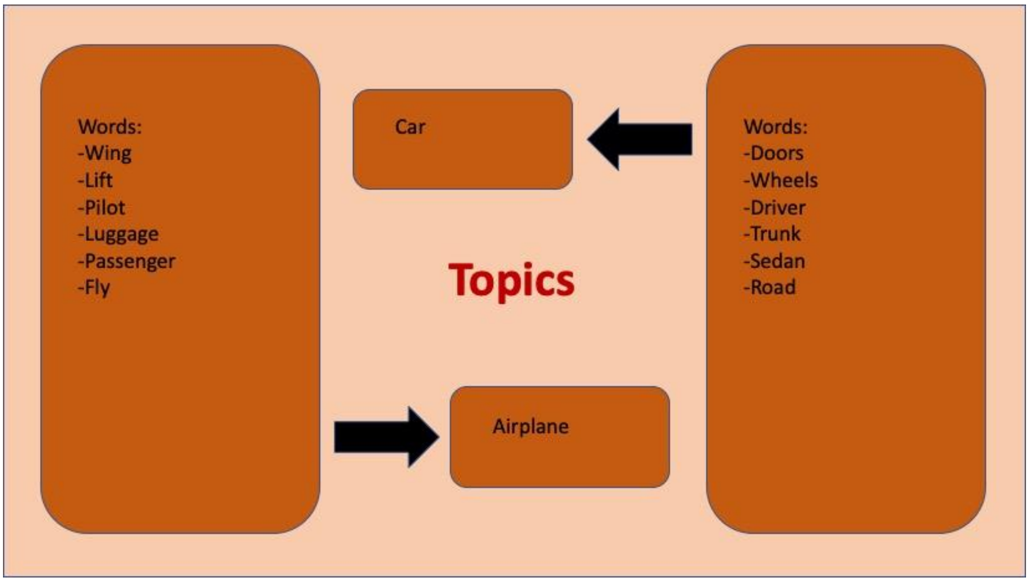


Figure 7.6: Inferring topics from word groupings

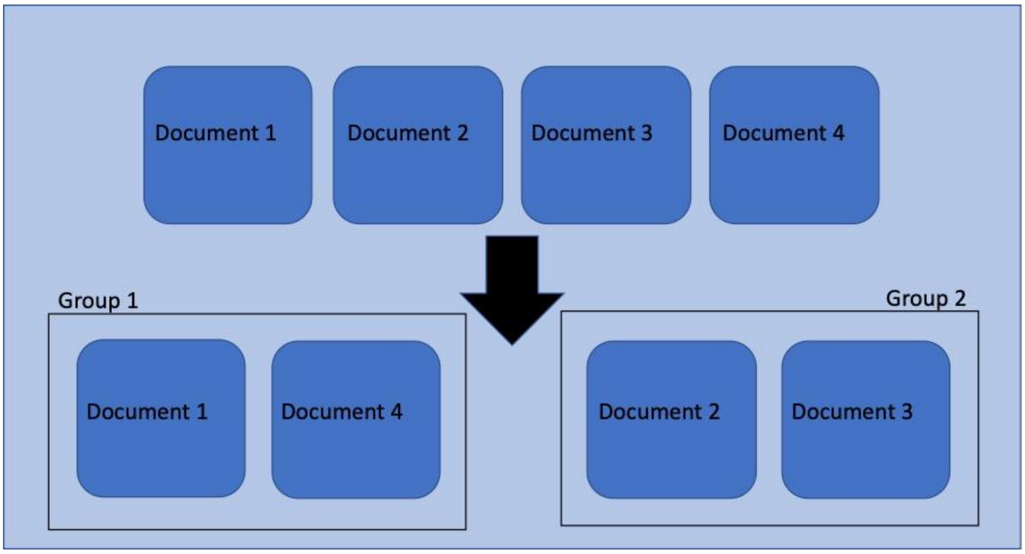


Figure 7.7: Sorting/categorizing documents

```
SHAPE:
(93239, 11)

COLUMN NAMES:
Index(['IDLink', 'Title', 'Headline', 'Source', 'Topic', 'PublishDate',
       'SentimentTitle', 'SentimentHeadline', 'Facebook', 'GooglePlus',
       'LinkedIn'],
      dtype='object')

HEAD:
    IDLink                                           Title  \
0  99248.0  Obama Lays Wreath at Arlington National Cemetery
1  10423.0       A Look at the Health of the Chinese Economy

                                            Headline     Source    Topic  \
0  Obama Lays Wreath at Arlington National Cemete...  USA TODAY    obama
1  Tim Haywood, investment director business-unit...  Bloomberg  economy

           PublishDate  SentimentTitle  SentimentHeadline  Facebook  \
0  2002-04-02 00:00:00        0.000000          -0.053300        -1
1  2008-09-20 00:00:00        0.208333          -0.156386        -1

   GooglePlus  LinkedIn
0          -1        -1
1          -1        -1
```

Figure 7.8: Raw data

```
HEADLINES:
['Obama Lays Wreath at Arlington National Cemetery. President Barack Obama has laid a wreath at the Tomb of the Unknowns to hon
or', 'Tim Haywood, investment director business-unit head for fixed income at Gam, discusses the China beige book and the state
of the economy.', "Nouriel Roubini, NYU professor and chairman at Roubini Global Economics, explains why the global economy is
n't facing the same conditions", "Finland's economy expanded marginally in the three months ended December, after contracting i
n the previous quarter, preliminary figures from Statistics Finland showed Monday. ", 'Tourism and public spending continued to
boost the economy in January, in light of contraction in private consumption and exports, according to the Bank of Thailand dat
a. ']

LENGTH:
93239
```

Figure 7.9: A list of headlines

```
Over 100 attendees expected to see latest version of Microsoft Dynamics SL and Dynamics GP (PRWeb February 29, 2016) Read the f
ull story at http://www.prweb.com/releases/2016/03/prweb13238571.htm
```

Figure 7.10: The fifth headline

```
DETECTED LANGUAGE:
en
```

Figure 7.11: Detected language

```
['Over', '100', 'attendees', 'expected', 'to', 'see', 'latest', 'version', 'of', 'Microsoft', 'Dynamics', 'SL', 'and', 'Dynamic
s', 'GP', '(PRWeb', 'February', '29,', '2016)', 'Read', 'the', 'full', 'story', 'at', 'http://www.prweb.com/releases/2016/03/pr
web13238571.htm', '']
```

Figure 7.12: String split using white spaces

```
['Over', '100', 'attendees', 'expected', 'to', 'see', 'latest', 'version', 'of', 'Microsoft', 'Dynamics', 'SL', 'and', 'Dynamic
s', 'GP', '(PRWeb', 'February', '29,', '2016)', 'Read', 'the', 'full', 'story', 'at', 'URL', '']
```

Figure 7.13: URLs replaced with the URL string

```
['Over', '100', 'attendees', 'expected', 'to', 'see', 'latest', 'version', 'of', 'Microsoft', 'Dynamics', 'SL', 'and', 'Dynamic
s', 'GP', 'PRWeb', 'February', '29', '2016', 'Read', 'the', 'full', 'story', 'at', 'URL', '']
```

Figure 7.14: Punctuation replaced with newline character

```
['Over', '', 'attendees', 'expected', 'to', 'see', 'latest', 'version', 'of', 'Microsoft', 'Dynamics', 'SL', 'and', 'Dynamics',
'GP', 'PRWeb', 'February', '', '', 'Read', 'the', 'full', 'story', 'at', 'URL', '']
```

Figure 7.15: Numbers replaced with empty strings

```
['over', '', 'attendees', 'expected', 'to', 'see', 'latest', 'version', 'of', 'microsoft', 'dynamics', 'sl', 'and', 'dynamics',
'gp', 'prweb', 'february', '', '', 'read', 'the', 'full', 'story', 'at', 'URL', '']
```

Figure 7.16: Uppercase letters converted to lowercase

```
['over', 'attendees', 'expected', 'to', 'see', 'latest', 'version', 'of', 'microsoft', 'dynamics', 'sl', 'and', 'dynamics', 'g
p', 'prweb', 'february', 'read', 'the', 'full', 'story', 'at']
```

Figure 7.17: String URL removed

```
['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', 'youre', 'youve', 'youll', 'youd', 'your', 'yours', 'yours
elf', 'yourselves', 'he', 'him', 'his', 'himself', 'she', 'shes', 'her', 'hers', 'herself', 'it', 'its', 'its', 'itself', 'the
y', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', 'thatll', 'these', 'those', 'am',
'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'th
e', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'betwee
n', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'ov
er', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each',
'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's',
't', 'can', 'will', 'just', 'don', 'dont', 'should', 'shouldve', 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', 'a
rent', 'couldn', 'couldnt', 'didn', 'didnt', 'doesn', 'doesnt', 'hadn', 'hadnt', 'hasn', 'hasnt', 'haven', 'havent', 'isn', 'is
nt', 'ma', 'mightn', 'mightnt', 'mustn', 'mustnt', 'needn', 'neednt', 'shan', 'shant', 'shouldn', 'shouldnt', 'wasn', 'wasnt',
'weren', 'werent', 'won', 'wont', 'wouldn', 'wouldnt']
```

Figure 7.18: List of stop words

```
['attendees', 'expected', 'see', 'latest', 'version', 'microsoft', 'dynamics', 'sl', 'dynamics', 'gp', 'prweb', 'february', 're
ad', 'full', 'story']
```

Figure 7.19: Stop words removed from the headline

```
['attendee', 'expect', 'see', 'latest', 'version', 'microsoft', 'dynamics', 'sl', 'dynamics', 'gp', 'prweb', 'february', 'rea
d', 'full', 'story']
```

Figure 7.20: Output after performing lemmatization

```
['attendee', 'expect', 'latest', 'version', 'microsoft', 'dynamics', 'dynamics', 'prweb', 'february', 'story']
```

Figure 7.21: Headline number five post-cleaning

```
HEADLINES:
[['obama', 'wreath', 'arlington', 'national', 'cemetery', 'president', 'barack', 'obama', 'wreath', 'unknown', 'honor'], ['hayw
ood', 'investment', 'director', 'businessunit', 'income', 'discus', 'china', 'beige', 'state', 'economy'], ['nouriel', 'roubin
i', 'professor', 'chairman', 'roubini', 'global', 'economics', 'explain', 'global', 'economy', 'facing', 'conditions'], ['finla
nd', 'economy', 'expand', 'marginally', 'three', 'month', 'december', 'contracting', 'previous', 'quarter', 'preliminary', 'fig
ure', 'statistics', 'finland', 'monday'], ['tourism', 'public', 'spending', 'continue', 'boost', 'economy', 'january', 'light',
'contraction', 'private', 'consumption', 'export', 'accord', 'thailand']]

LENGTH:
92948
```

Figure 7.22: Headline and its length

```
['obama wreath arlington national cemetery president barack obama wreath unknown honor', 'haywood investment director businessu
nit income discus china beige state economy', 'nouriel roubini professor chairman roubini global economics explain global econo
my facing conditions', 'finland economy expand marginally three month december contracting previous quarter preliminary figure
statistics finland monday', 'tourism public spending continue boost economy january light contraction private consumption expor
t accord thailand', 'attendee expect latest version microsoft dynamics dynamics prweb february story', 'ramallah february pales
tine liberation organization sectretarygeneral erekat thursday express concern kenyan president uhuru kenyattas visit jerusalem
jordan valley', 'first michelle obama speak state white house washington wednesday interactive student workshop musical legacy
charles student school community organization across country participate quotin performance white housequot series', 'hancock c
ounty early monday morning family years', 'delhi feb29 technology giant microsoft target rival apple series focusing windows gr
oss windows machine']
```

Figure 7.23: Headlines cleaned for modeling

```
['running shoes extra', 'class crunch intense workout pulley system', 'thousand natural product', 'natural product ex
plore beauty supplement', 'fitness weekend south beach spark activity', 'kayla harrison sacrifice', 'sonic treatment
alzheimers disease', 'ultrasound brain restore memory alzheimers needle onlyso farin mouse', 'apple researchkit reall
y medical research', 'warning chantix drink taking might remember']
```

Figure 7.24: Tweets cleaned for modeling

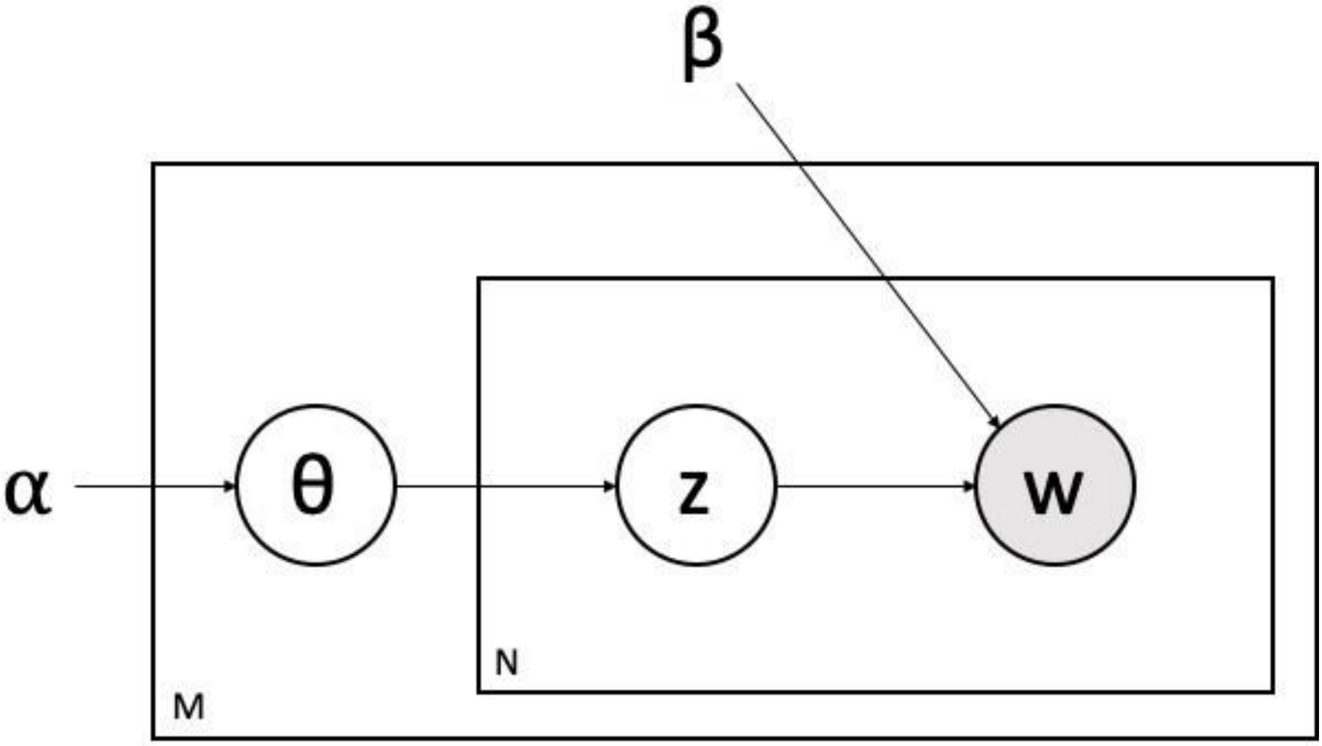


Figure 7.25: Graphical representation of LDA

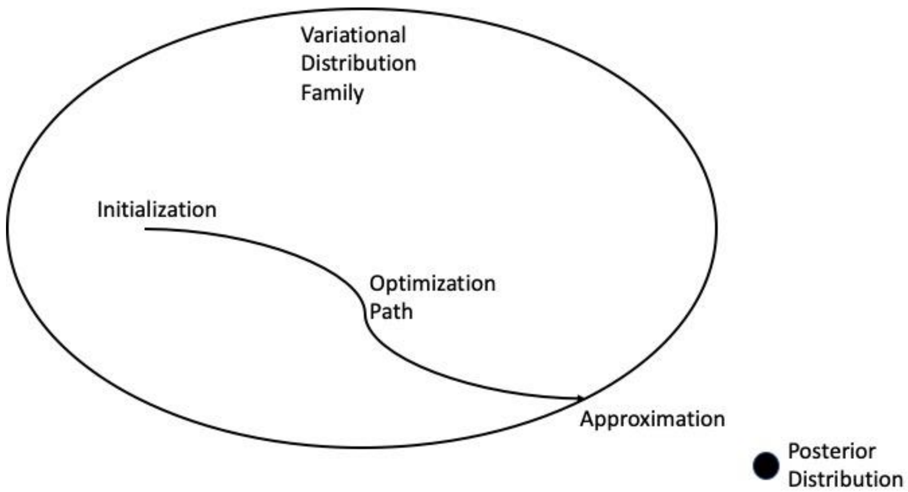


Figure 7.26: The variational inference process

$$\boxed{q(\theta, z|\gamma, \phi) = q(\theta|\gamma) \prod_{n=1}^{N} q(z_n|\phi_n)}$$

Figure 7.27: Formula for the family of distributions, q

(0, 407)	1
(0, 88)	1
(0, 643)	1
(0, 557)	1
(0, 572)	2

Figure 7.28: The bag-of-words data structure

$$PP = \hat{P}(w_1, \ldots, w_m)^{-1/m}$$

Figure 7.29: Formula of perplexity

	Number Of Topics	Perplexity Score
0	1	510.011710
1	2	464.310162
2	3	413.054650
3	4	431.545934
4	6	511.728157
5	8	542.678576
6	10	572.124718

Figure 7.30: Data frame containing number of topics and perplexity score

```
<matplotlib.axes._subplots.AxesSubplot at 0x1ebe05b15f8>
```

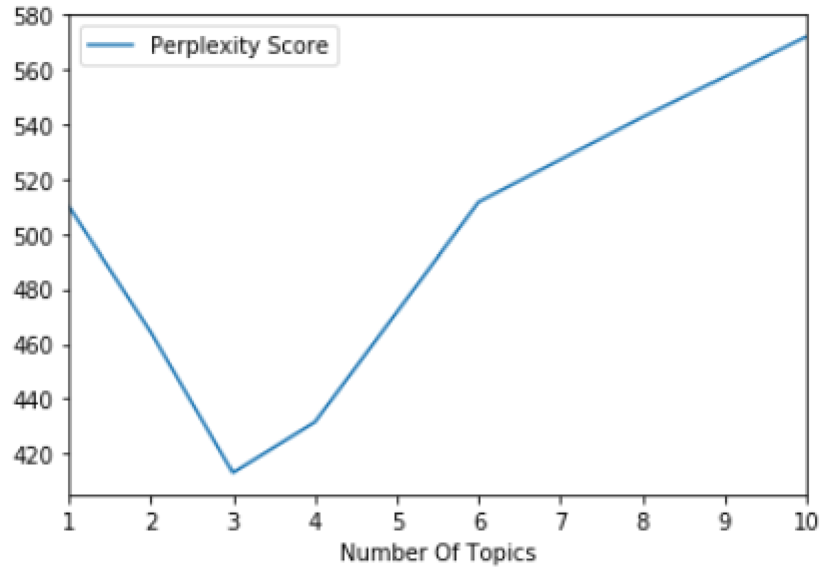


Figure 7.31: Line plot view of perplexity as a function of the number of topics

```
LatentDirichletAllocation(batch_size=128, doc_topic_prior=None,
             evaluate_every=-1, learning_decay=0.7,
             learning_method='online', learning_offset=10.0,
             max_doc_update_iter=100, max_iter=10, mean_change_tol=0.001,
             n_components=3, n_jobs=None, n_topics=None, perp_tol=0.1,
             random_state=0, topic_word_prior=None,
             total_samples=1000000.0, verbose=0)
```

Figure 7.32: LDA model

```
(92948, 3)
[[0.90423071 0.04761949 0.0481498 ]
 [0.0449056  0.04292327 0.91217113]
 [0.0435693  0.0441942  0.91223649]
 ...
 [0.20977116 0.03942095 0.75080789]
 [0.09239268 0.07121637 0.83639094]
 [0.20062764 0.41458136 0.384791   ]]
```

Figure 7.33: Topic-document matrix and its dimensions

```
(3, 1000)
[[3.67812459e-01 3.83046413e-01 3.79939561e-01 ... 3.48448881e-01
  1.18665576e+02 4.62012727e+02]
 [3.36269915e-01 2.72144107e+02 2.61257455e+01 ... 3.35946774e-01
  2.05558903e+02 3.94048139e-01]
 [2.74795972e+02 4.27720110e-01 1.89390109e+02 ... 2.31713244e+02
  1.79236579e+02 4.10569467e-01]]
```

Figure 7.34: Word-topic matrix and its dimensions

	Topic0	Topic1	Topic2
Word0	(0.1009, obama)	(0.1025, microsoft)	(0.0874, economy)
Word1	(0.0874, president)	(0.0235, windows)	(0.0301, economic)
Word2	(0.0502, barack)	(0.0229, company)	(0.0161, palestine)
Word3	(0.0157, obamas)	(0.0185, microsofts)	(0.0152, growth)
Word4	(0.015, washington)	(0.0155, announce)	(0.0129, global)
Word5	(0.014, state)	(0.014, today)	(0.0126, palestinian)
Word6	(0.013, house)	(0.0105, release)	(0.011, government)
Word7	(0.0119, white)	(0.0088, business)	(0.0103, minister)
Word8	(0.0117, administration)	(0.0088, update)	(0.0101, world)
Word9	(0.0087, visit)	(0.0075, surface)	(0.01, china)

Figure 7.35: Word-topic table

```
                                                  Topic0  \
Doc0  (0.9776, March 13 marked the 75th anniversary ...
Doc1  (0.9776, Preying on the minds of financial mar...
Doc2  (0.9772, Obama has narrowed his list to 3 nomi...
Doc3  (0.9768, Member nations of the Organization of...
Doc4  (0.9767, Malia Obama is 17 and probably wants ...
Doc5  (0.9765, Democratic presidential front-runner ...
Doc6  (0.9758, Chinese Premier Li Keqiang pledged th...
Doc7  (0.9756, UNITED NATIONS """ France said Friday...
Doc8  (0.9756, French Foreign Minister Laurent Fabiu...
Doc9  (0.9755, KANSAS CITY """ Missouri Republican a...

                                                  Topic1  \
Doc0  (0.9776, That appears to be the thinking behin...
Doc1  (0.9764, Arundhati Bhattacharya recognises tha...
Doc2  (0.9757, France's fragile economy has cooled i...
Doc3  (0.9755, Software maker Microsoft Corp is sell...
Doc4  (0.9755, WASHINGTON (AP) — President Barack Ob...
Doc5  (0.9754, France's Palestine peace plan is part...
Doc6  (0.9752, Patent trolls drain $1.5 billion a we...
Doc7  (0.9751, Rancho Mirage, California (CNN) Presi...
Doc8  (0.975, 2 economy could be sucked into a Japan...
Doc9  (0.975, Economist with the University of Ghana...

                                                  Topic2
Doc0  (0.9783, President Barack Obama drinks water a...
Doc1  (0.9781, Ifo economist Klaus Wohlrabe told Reu...
Doc2  (0.978, Microsoft's latest Windows Phone, the ...
Doc3  (0.978, Microsoft CEO Satya Nadella discussed ...
Doc4  (0.9779, People’s Bank of China Governor Zhou ...
Doc5  (0.9778, President Obama welcomed the Super Bo...
Doc6  (0.9778, The UK is facing a digital skills cri...
Doc7  (0.9778, Microsoft said Monday that it is buyi...
Doc8  (0.9778, Microsoft has been on the acquisition...
Doc9  (0.9777, Twitter, Microsoft, Facebook and YouT...
```

Figure 7.36: Topic-document table

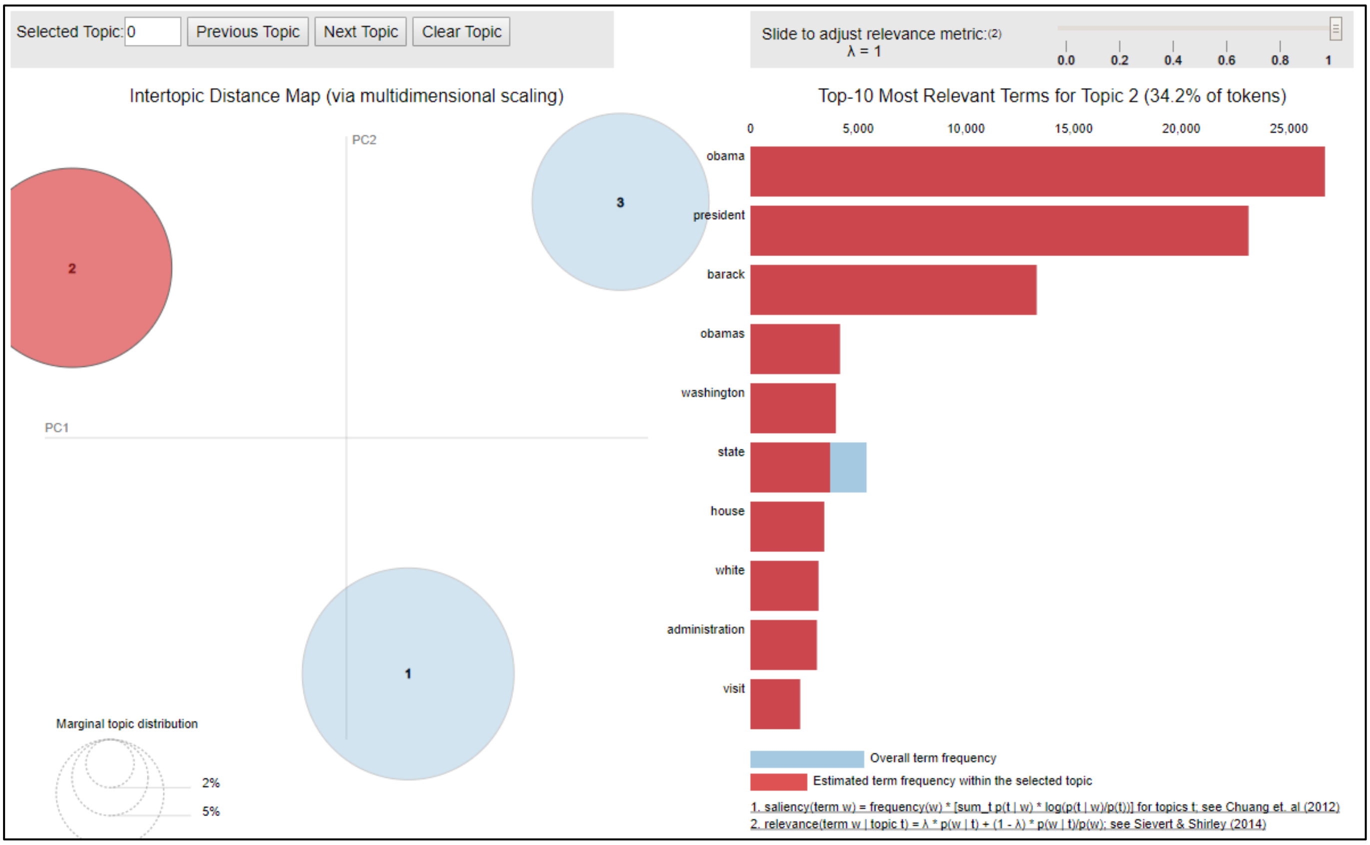


Figure 7.37: A histogram and biplot for the LDA model

```
LENGTH:
56455

COUNTS:
[[     0 18477]
 [     1 15766]
 [     2 22212]]
```

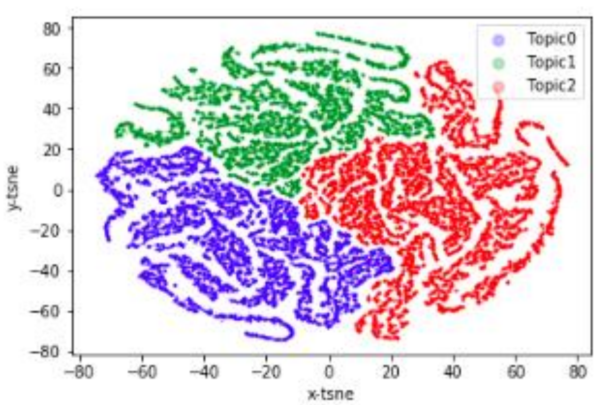


Figure 7.38: t-SNE plot with metrics around the distribution of the topics across the corpus

```
LatentDirichletAllocation(batch_size=128, doc_topic_prior=None,
             evaluate_every=-1, learning_decay=0.7,
             learning_method='online', learning_offset=10.0,
             max_doc_update_iter=100, max_iter=10, mean_change_tol=0.001,
             n_components=4, n_jobs=None, n_topics=None, perp_tol=0.1,
             random_state=0, topic_word_prior=None,
             total_samples=1000000.0, verbose=0)
```

Figure 7.39: LDA model

```
                      Topic0                        Topic1                 Topic2  \
Word0    (0.0344, palestine)           (0.1332, obama)      (0.1062, economy)
Word1   (0.0283, washington)       (0.1155, president)     (0.0365, economic)
Word2  (0.0269, palestinian)          (0.0664, barack)       (0.0185, growth)
Word3        (0.0244, house)          (0.0208, obamas)         (0.017, world)
Word4        (0.0225, white)  (0.0154, administration)       (0.0157, global)
Word5      (0.0214, tuesday)           (0.0122, state)     (0.0126, minister)
Word6       (0.0185, people)           (0.0107, trump)        (0.0122, china)
Word7     (0.0162, american)      (0.0102, republican)      (0.0114, percent)
Word8        (0.0149, unite)           (0.0087, union)   (0.0106, government)
Word9        (0.0146, state)           (0.0087, visit)       (0.0103, market)

                     Topic3
Word0   (0.1155, microsoft)
Word1     (0.0265, windows)
Word2     (0.0259, company)
Word3  (0.0209, microsofts)
Word4    (0.0171, announce)
Word5       (0.0158, today)
Word6     (0.0115, release)
Word7      (0.0099, update)
Word8    (0.0091, business)
Word9     (0.0084, surface)
```

Figure 7.40: The word-topic table using the four-topic LDA model

```
                                                       Topic0  \
Doc0   (0.9618, President Barack Obama on Friday will...
Doc1   (0.9494, NEW YORK (Reuters) - Facing a hostile...
Doc2   (0.9459, The Personalization Gallery offers a ...
Doc3   (0.9459, In the budget he plans to release tom...
Doc4   (0.9458, When Microsoft introduced its new on-...
Doc5   (0.9369, President Barack Obama speaks at the ...
Doc6   (0.9369, In an email interview, Adam Fforde, p...
Doc7   (0.9358, A panel of Fox Business pundits excor...
Doc8   (0.9317, President Obama talked about efforts ...
Doc9   (0.9315, An Israeli soldier takes aim during c...

                                                       Topic1  \
Doc0   (0.9686, Artists including Missy Elliott, Kell...
Doc1   (0.9686, President Barack Obama has chosen a n...
Doc2   (0.9686, WASHINGTON — President Barack Obama h...
Doc3   (0.9671, Plouffe, who managed President Obama'...
Doc4   (0.9671,  is dragging the economy through the ...
Doc5   (0.967, President Obama challenged the content...
Doc6   (0.9578, While many people opt for social medi...
Doc7   (0.9558, WASHINGTON """ President Barack Obama...
Doc8   (0.9558, KIEV, March 16. /TASS/. Overwhelming ...
Doc9   (0.9557, Premier Li Keqiang said Wednesday tha...

                                                       Topic2  \
Doc0   (0.9739, WASHINGTON """ President Obama on Sat...
Doc1   (0.9739, TULKARM, November 29, 2015 (WAFA) """...
Doc2   (0.9729, Chris Christie boasted that he banned...
Doc3   (0.9729, CHANTILLY, Va. """ President Barack O...
Doc4   (0.9728, As Microsoft's mobile platform contin...
Doc5   (0.9714, Its growth estimate for 2015-16 has j...
Doc6   (0.9709, The rivalry between Sony and Microsof...
Doc7   (0.9706, Kuwait's economy contracted last year...
Doc8   (0.9706, Kuwait's economy contracted last year...
Doc9   (0.9698, The economic situation in Europe is l...

                                                       Topic3
Doc0   (0.9749, US President Barack Obama has been at...
Doc1   (0.9729, US President Barack Obama on Wednesda...
Doc2   (0.9721, 2 economy could be sucked into a Japa...
Doc3   (0.9721, """When I began working on this conce...
Doc4   (0.9721, The Estonian economy was also positiv...
Doc5   (0.9718, Japan's surprise, albeit modest, meas...
Doc6   (0.9711, The importance of a first good impres...
Doc7   (0.971, The Obama administration on Friday ask...
Doc8   (0.9698, REDMOND, Wash., Dec. 9, 2015 /PRNewsw...
Doc9   (0.9696, The controversial move by Brazil's pr...
```

Figure 7.41: The document-topic table using the four-topic LDA model

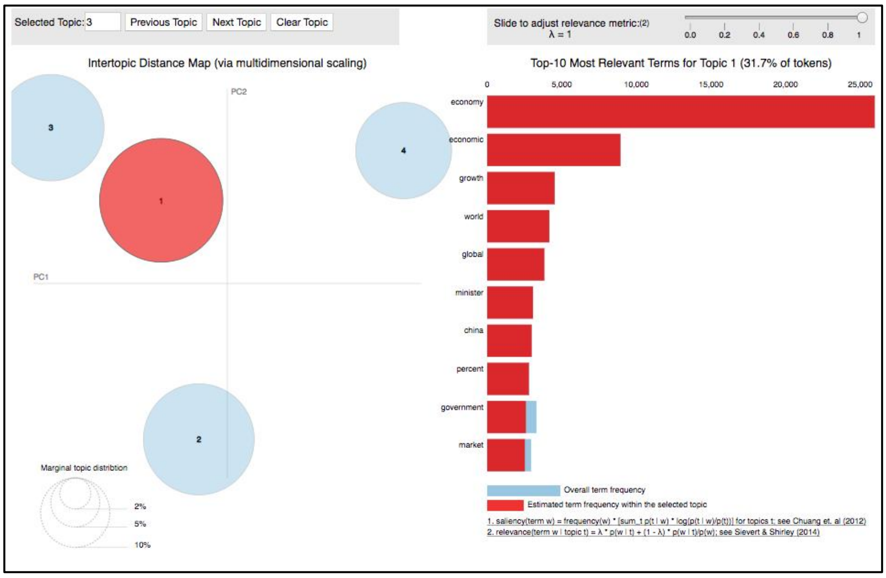


Figure 7.42: A histogram and biplot describing the four-topic LDA model

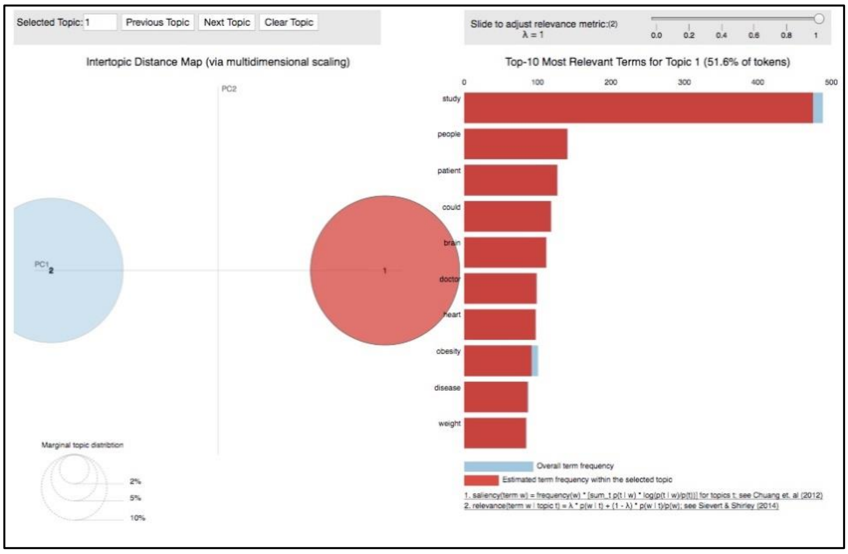


Figure 7.43: A histogram and biplot for the LDA model trained on health tweets

```
(0, 572)        0.4507592105469774
(0, 557)        0.4666029379072775
(0, 643)        0.2348310160024775
(0, 88)         0.2807099986206347
(0, 407)        0.667198713308869
```

Figure 7.44: Output of the TF-IDF vectorizer

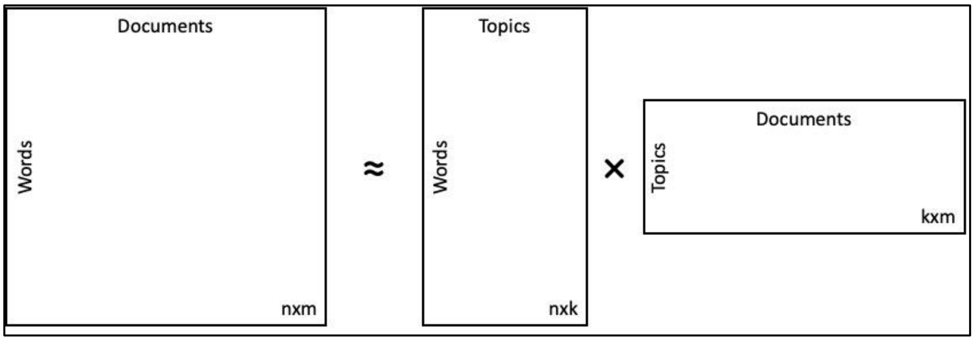


Figure 7.45: The matrix factorization

$$H^{i+1} \leftarrow H^{i} \frac{(W^{i})^{T} X}{(W^{i})^{T} W^{i} H^{i}}$$

Figure 7.46: First update rule

$$w^{i+1} \leftarrow w^{i} \frac{X\,(H^{i+1})^{T}}{W^{i}H^{i+1}\,(H^{i+1})^{T}}$$

Figure 7.47: Second update rule

```
NMF(alpha=0.1, beta_loss='frobenius', init='nndsvda', l1_ratio=0.5,
  max_iter=200, n_components=4, random_state=0, shuffle=False, solver='mu',
  tol=0.0001, verbose=0)
```

Figure 7.48: Defining the NMF model

	Topic0	Topic1	Topic2
Word0	(0.0696, obama)	(0.0628, economy)	(0.0869, microsoft)
Word1	(0.0646, president)	(0.0212, economic)	(0.0306, windows)
Word2	(0.0484, barack)	(0.0179, growth)	(0.0196, company)
Word3	(0.0157, washington)	(0.0144, global)	(0.0162, announce)
Word4	(0.0149, house)	(0.0128, china)	(0.0124, microsofts)
Word5	(0.0144, white)	(0.0111, percent)	(0.0118, update)
Word6	(0.0127, obamas)	(0.0109, world)	(0.0106, release)
Word7	(0.0109, state)	(0.0097, quarter)	(0.01, today)
Word8	(0.0096, administration)	(0.0093, market)	(0.0096, surface)
Word9	(0.0081, first)	(0.0086, country)	(0.0085, cloud)

	Topic3
Word0	(0.0881, palestine)
Word1	(0.0766, palestinian)
Word2	(0.0309, israeli)
Word3	(0.0278, israel)
Word4	(0.0172, state)
Word5	(0.0094, international)
Word6	(0.0092, ramallah)
Word7	(0.0089, minister)
Word8	(0.0079, unite)
Word9	(0.0078, force)

Figure 7.49: The word-topic table containing probabilities

```
                                                    Topic0  \
Doc0  (0.0844, NCRI - The Iranian regime's former Mi...
Doc1  (0.0844, South Africa's economy shrank sharply...
Doc2  (0.0844, Horacio Gutierrez, Microsoft's genera...
Doc3  (0.0844, A Microsoft recruiting event at the U...
Doc4  (0.0844, The Federal Reserve's recent rate hik...
Doc5  (0.0844, President Barack Obama received a chi...
Doc6  (0.0844, President Obama met with gun control ...
Doc7  (0.0844, (CNN) """Leaders gathered in Paris to...
Doc8  (0.0844, Fears have returned that China's debt...
Doc9  (0.0844, Russia is not ready to share US Presi...

                                                    Topic1  \
Doc0  (0.0677, Both China's central bank and a respe...
Doc1  (0.0677, TAMPA -- Sen. Marco Rubio (R-Fla.) sa...
Doc2  (0.0677, WASHINGTON - President Barack Obama i...
Doc3  (0.0677, The U.S. Supreme Court on Friday agre...
Doc4  (0.0677, WASHINGTON"""President Barack Obama w...
Doc5  (0.0677, One of the challenges for writing app...
Doc6  (0.0677, President Barack Obama speaks during ...
Doc7        (0.0677, The U.S. economy is humming again. )
Doc8  (0.0677, WASHINGTON — President Barack Obama s...
Doc9  (0.0677, Microsoft to shut down portal site MS...

                                                    Topic2  \
Doc0  (0.0836, Colin Fenton, managing partner at Bla...
Doc1  (0.0836, As I study in Canada, I am exposed to...
Doc2  (0.0836, Ban Ki Mun: Sramota me zbog Izraela i...
Doc3  (0.0836, But the argument that Microsoft is wi...
Doc4  (0.0836, 6:55 p.m. On the heels of Donald Trum...
Doc5  (0.0836, Colorado's economy had the fourth str...
Doc6  (0.0836, Back in September 2014 I wrote an art...
Doc7  (0.0836, During its developer conference, Micr...
Doc8  (0.0836, Speaking in Japan after a summit with...
Doc9  (0.0836, Korea's exports marked the worst slum...

                                                    Topic3
Doc0  (0.1078, SYDNEY, April 18 (Xinhua) -- Australi...
Doc1  (0.1078, It's both fascinating and ironic how ...
Doc2  (0.0856, German government spending on refugee...
Doc3  (0.0852, President Barack Obama, center, walks...
Doc4  (0.0842, The gig economy tends to divide opini...
Doc5  (0.0828, I hope the Committee on the Future Ec...
Doc6  (0.0815, President Obama has spent the last se...
Doc7  (0.0815, OTTAWA -- Barack Obama has arrived in...
Doc8  (0.0815, For Max Wolff, chief economist at Man...
Doc9  (0.0815, Just over a year ago, Microsoft annou...
```

Figure 7.50: The document-topic table containing probabilities

```
(92948, 4)
[[5.12543656e-02 3.63195740e-15 3.10455307e-34 7.82654193e-16]
 [7.41162473e-04 2.04135415e-02 6.83519643e-15 2.13620923e-03]
 [2.96652472e-15 1.94116773e-02 4.78856726e-21 1.20646716e-18]
 ...
 [9.58970155e-06 3.41045363e-03 6.15591120e-04 3.23909905e-02]
 [6.37006094e-07 1.31884850e-07 3.39453370e-08 6.14080053e-02]
 [4.46386338e-05 1.15780717e-04 1.84769162e-02 2.00666640e-03]]
```

Figure 7.51: Shape and example of data

```
LENGTH:
92946

COUNTS:
[[     0 28977]
 [     1 32946]
 [     2 22146]
 [     3  8877]]
```

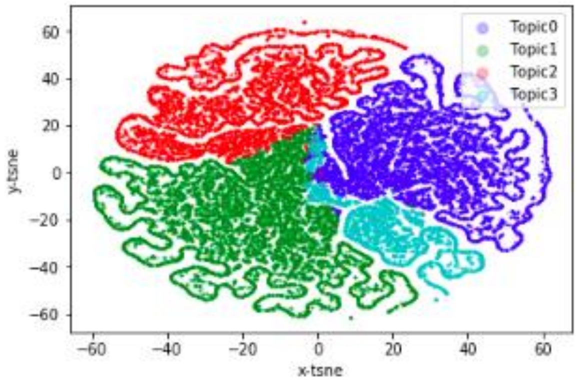


Figure 7.52: t-SNE plot with metrics summarizing the topic distribution across the corpus

	Topic0	Topic1
Word0	(0.3726, study)	(0.5974, latfit)
Word1	(0.0259, cancer)	(0.0477, steps)
Word2	(0.0208, people)	(0.0448, today)
Word3	(0.0185, health)	(0.0404, exercise)
Word4	(0.0184, obesity)	(0.0274, healthtips)
Word5	(0.0182, brain)	(0.0257, workout)
Word6	(0.0173, suggest)	(0.0204, getting)
Word7	(0.0167, weight)	(0.0193, fitness)
Word8	(0.0159, woman)	(0.0143, great)
Word9	(0.0131, death)	(0.0132, morning)

Figure 7.53: The word-topic table with probabilities

Lesson 8: Market Basket Analysis

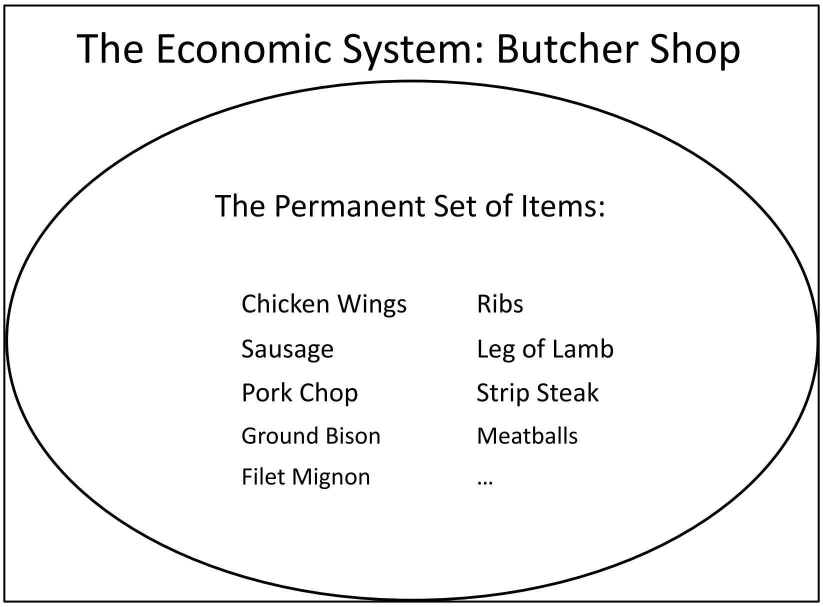


Figure 8.1: An example market basket where the economic system is the butcher shop and the permanent set of items is all the meat products offered by the butcher

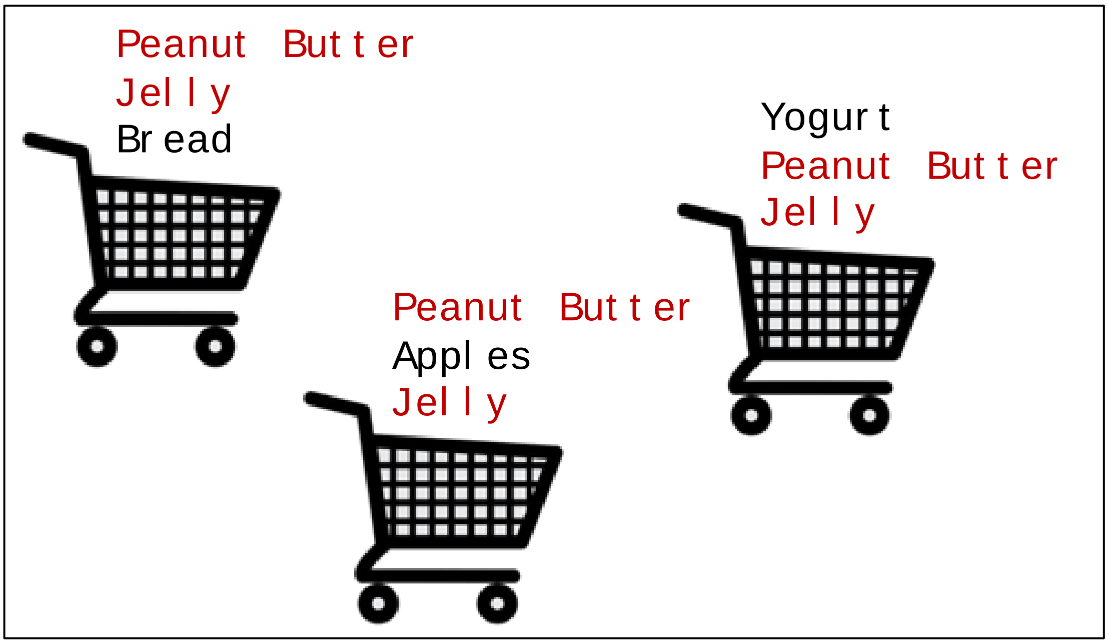


Figure 8.2: A visualization of market basket analysis

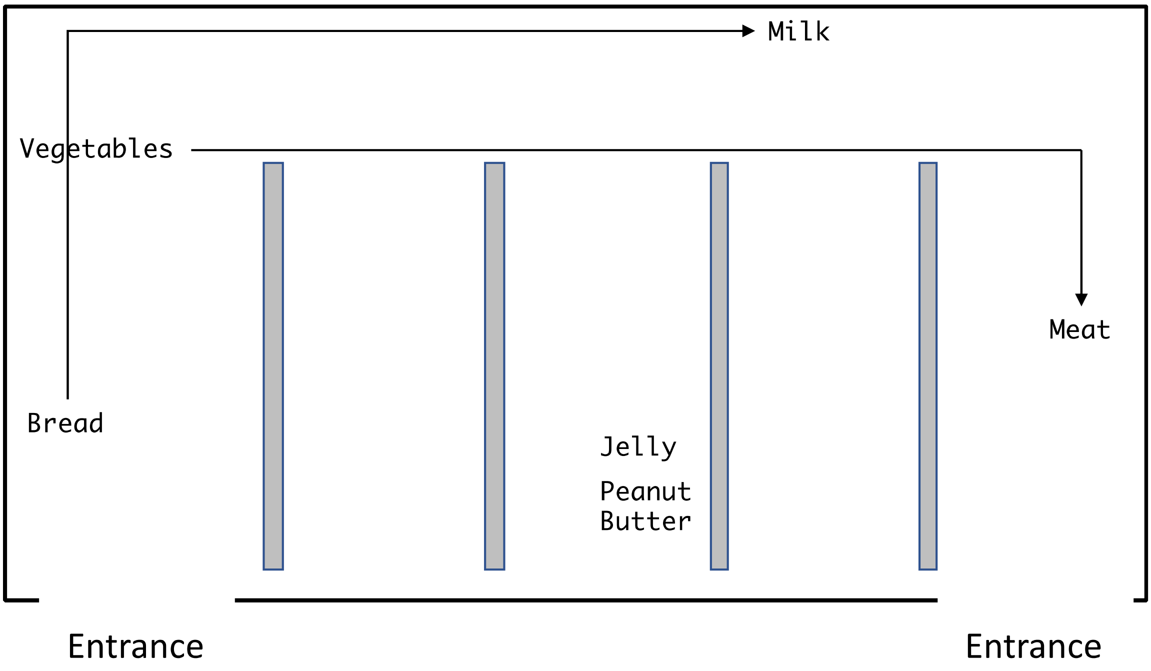


Figure 8.3: How product associations can help inform efficient and lucrative store layouts

$$Support(X \Rightarrow Y) = Support(X,Y) = P(X,Y) = \frac{Frequency(X,Y)}{N}$$

Figure 8.4: Formula for support

Figure 8.5: Formula for confidence

$$Lift(X \Rightarrow Y) = \frac{Support(X,Y)}{Support(X) * Support(Y)} = \frac{P(X,Y)}{P(X) * P(Y)}$$

Figure 8.6: Formula for lift

$$Leverage(X \Rightarrow Y) = Support(X,Y) - \big(Support(X) * Support(Y)\big) = P(X,Y) - \big(P(X) * P(Y)\big)$$

Figure 8.7: Formula for leverage

$$Conviction(X \Rightarrow Y) = \frac{1 - Support(Y)}{1 - Confidence(X \Rightarrow Y)}$$

Figure 8.8: Formula for conviction

```
N = 10
Freq(x) = 7
Freq(y) = 5
Freq(x, y) = 4
```

Figure 8.9: Screenshot of the frequencies

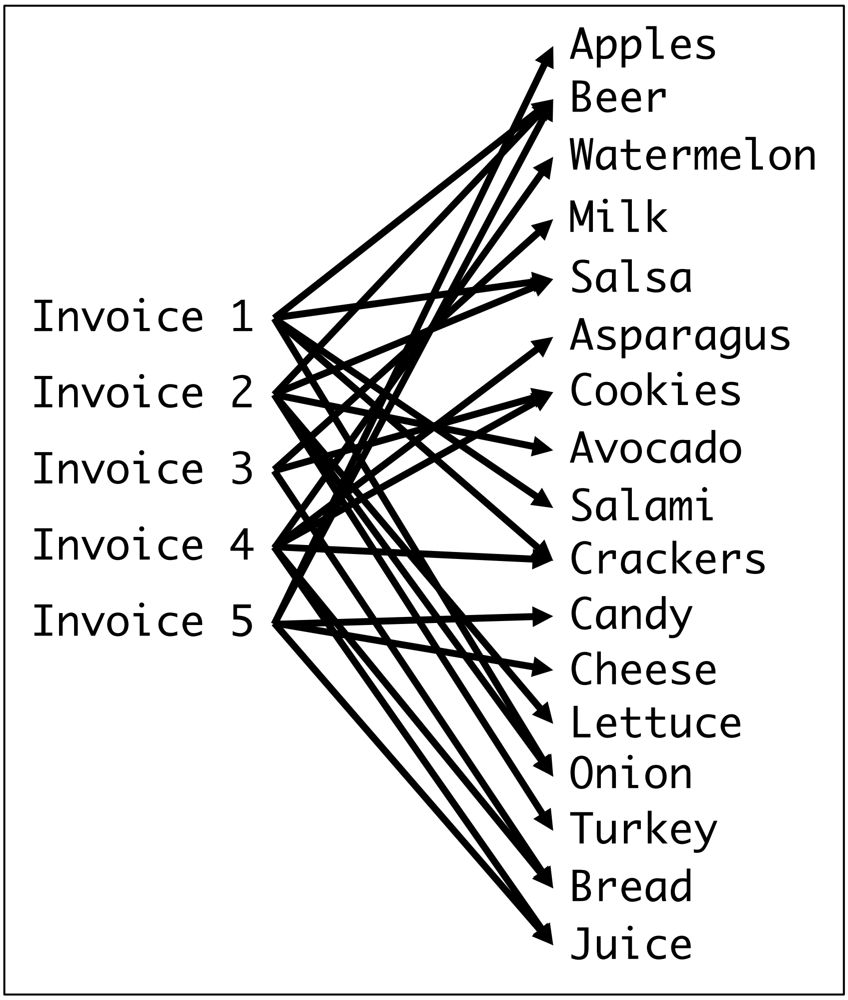


Figure 8.10: Each available product is going to map back to multiple invoice numbers

	InvoiceNo	StockCode	Description	Quantity	InvoiceDate	UnitPrice	CustomerID	Country
0	536365	85123A	WHITE HANGING HEART T-LIGHT HOLDER	6	2010-12-01 08:26:00	2.55	17850.0	United Kingdom
1	536365	71053	WHITE METAL LANTERN	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
2	536365	84406B	CREAM CUPID HEARTS COAT HANGER	8	2010-12-01 08:26:00	2.75	17850.0	United Kingdom
3	536365	84029G	KNITTED UNION FLAG HOT WATER BOTTLE	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
4	536365	84029E	RED WOOLLY HOTTIE WHITE HEART.	6	2010-12-01 08:26:00	3.39	17850.0	United Kingdom
5	536365	22752	SET 7 BABUSHKA NESTING BOXES	2	2010-12-01 08:26:00	7.65	17850.0	United Kingdom
6	536365	21730	GLASS STAR FROSTED T-LIGHT HOLDER	6	2010-12-01 08:26:00	4.25	17850.0	United Kingdom
7	536366	22633	HAND WARMER UNION JACK	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
8	536366	22632	HAND WARMER RED POLKA DOT	6	2010-12-01 08:28:00	1.85	17850.0	United Kingdom
9	536367	84879	ASSORTED COLOUR BIRD ORNAMENT	32	2010-12-01 08:34:00	1.69	13047.0	United Kingdom

Figure 8.11: The raw online retail data

```
InvoiceNo              object
StockCode              object
Description            object
Quantity                int64
InvoiceDate    datetime64[ns]
UnitPrice             float64
CustomerID            float64
Country                object
dtype: object
```

Figure 8.12: Data type for each column in the dataset

	InvoiceNo	Description
0	536365	WHITE HANGING HEART T-LIGHT HOLDER
1	536365	WHITE METAL LANTERN
2	536365	CREAM CUPID HEARTS COAT HANGER
3	536365	KNITTED UNION FLAG HOT WATER BOTTLE
4	536365	RED WOOLLY HOTTIE WHITE HEART.
5	536365	SET 7 BABUSHKA NESTING BOXES
6	536365	GLASS STAR FROSTED T-LIGHT HOLDER
7	536366	HAND WARMER UNION JACK
8	536366	HAND WARMER RED POLKA DOT
9	536367	ASSORTED COLOUR BIRD ORNAMENT

Figure 8.13: The cleaned online retail dataset

	InvoiceNo	Description
229435	557056	SET OF 4 KNICK KNACK TINS DOILEY
229436	557057	RED POLKADOT BEAKER
229437	557057	BLUE POLKADOT BEAKER
229438	557057	DAIRY MAID TOASTRACK
229439	557057	BLUE EGG SPOON
229440	557057	RED EGG SPOON
229441	557057	MODERN FLORAL STATIONERY SET
229442	557057	FLORAL FOLK STATIONERY SET
229443	557057	CERAMIC BOWL WITH LOVE HEART DESIGN
229444	557057	WOOD STAMP SET THANK YOU

Figure 8.14: The cleaned dataset with only 5,000 unique invoice numbers

```
[['RED POLKADOT BEAKER ', 'BLUE POLKADOT BEAKER ', 'DAIRY MAID TOASTRACK', 'BLUE EGG  SPOON', 'RED  EGG  SPOON', 'MODERN FLORAL
STATIONERY SET', 'FLORAL FOLK STATIONERY SET', 'CERAMIC BOWL WITH LOVE HEART DESIGN', 'WOOD STAMP SET THANK YOU', 'WOOD STAMP S
ET HAPPY BIRTHDAY', 'PENS ASSORTED SPACEBALL', 'PENS ASSORTED FUNNY FACE', 'PENS ASSORTED FUNKY JEWELED ', 'SCOTTIE DOGS BABY B
IB', 'CHARLIE AND LOLA TABLE TINS', 'CHARLIE & LOLA WASTEPAPER BIN FLORA', 'CHARLIE & LOLA WASTEPAPER BIN BLUE', 'CHARLIE AND L
OLA FIGURES TINS', 'TV DINNER TRAY DOLLY GIRL', 'SET/20 RED RETROSPOT PAPER NAPKINS ', 'MINT KITCHEN SCALES', 'RED KITCHEN SCAL
ES', '36 FOIL HEART CAKE CASES', '36 FOIL STAR CAKE CASES ', 'ILLUSTRATED CAT BOWL ', 'POTTING SHED TEA MUG', 'CERAMIC STRAWBER
RY DESIGN MUG', 'RED RETROSPOT SHOPPER BAG', 'BUTTON BOX ', 'MINI CAKE STAND  HANGING STRAWBERY', 'LUNCH BAG DOILEY PATTERN ',
'JUMBO BAG STRAWBERRY', 'STRAWBERRY SHOPPER BAG', 'SUKI  SHOULDER BAG', 'JUMBO BAG ALPHABET', 'SKULL SHOULDER BAG', 'LUNCH BAG
BLACK SKULL.', 'TRADITIONAL WOODEN CATCH CUP GAME ', '10 COLOUR SPACEBOY PEN', 'JUMBO BAG SPACEBOY DESIGN', 'LUNCH BAG SPACEBOY
DESIGN ', "CHILDREN'S APRON DOLLY GIRL ", 'LUNCH BAG DOLLY GIRL DESIGN', 'TEATIME ROUND PENCIL SHARPENER ', 'SILVER HEARTS TABL
E DECORATION', 'PARISIENNE KEY CABINET ', 'PARISIENNE JEWELLERY DRAWER ', 'BUNDLE OF 3 SCHOOL EXERCISE BOOKS  ', 'JUMBO BAG DOI
LEY PATTERNS', 'DOILEY STORAGE TIN', 'SET OF 4 KNICK KNACK TINS POPPIES', 'SET OF 4 KNICK KNACK TINS DOILEY ', 'SET OF 3 REGENC
Y CAKE TINS', 'SET OF 3 WOODEN HEART DECORATIONS', 'SPACEBOY CHILDRENS BOWL', 'DOLLY GIRL CHILDRENS CUP', 'DOLLY GIRL CHILDRENS
BOWL', 'SPACE BOY CHILDRENS CUP', 'GARDENERS KNEELING PAD CUP OF TEA ', 'GARDENERS KNEELING PAD KEEP CALM ', 'CARTOON  PENCIL S
HARPENERS', 'POPART RECT PENCIL SHARPENER ASST', 'PIECE OF CAMO STATIONERY SET', 'POPART WOODEN PENCILS ASST', 'ORIGAMI VANILLA
INCENSE/CANDLE SET ', 'ORIGAMI JASMINE INCENSE/CANDLE SET', 'FRENCH FLORAL CUSHION COVER ', 'FRENCH LATTICE CUSHION COVER '],
['SET OF TEA COFFEE SUGAR TINS PANTRY', 'SET OF 3 CAKE TINS PANTRY DESIGN '], ['JUMBO BAG PINK VINTAGE PAISLEY', 'JUMBO  BAG BA
ROQUE BLACK WHITE', 'RIBBON REEL STRIPES DESIGN ', 'RIBBON REEL LACE DESIGN ', 'RIBBON REEL POLKADOTS ', 'TRAVEL CARD WALLET TR
ANSPORT', 'TRAVEL CARD WALLET FLOWER MEADOW', 'TRAVEL CARD WALLET VINTAGE LEAF', 'TRAVEL CARD WALLET VINTAGE TICKET', 'VINTAGE
2 METER FOLDING RULER', 'IVORY WICKER HEART LARGE', 'BUNDLE OF 3 ALPHABET EXERCISE BOOKS', 'BUNDLE OF 3 RETRO NOTE BOOKS', '20
DOLLY PEGS RETROSPOT', 'CLOTHES PEGS RETROSPOT PACK 24 ', 'VICTORIAN  METAL POSTCARD SPRING', 'ROLL WRAP VINTAGE CHRISTMAS', 'R
OLL WRAP VINTAGE SPOT ', 'ENAMEL MEASURING JUG CREAM', 'JUMBO BAG VINTAGE CHRISTMAS ', "JUMBO BAG 50'S CHRISTMAS ", 'SET OF 4 K
NICK KNACK TINS DOILY ', 'SET OF 4 KNICK KNACK TINS POPPIES', 'IVORY WICKER HEART LARGE', 'JINGLE BELL HEART ANTIQUE GOLD', 'SE
T OF 4 NAPKIN CHARMS CUTLERY', 'SET OF 4 NAPKIN CHARMS HEARTS', 'SET OF 4 KNICK KNACK TINS LEAF', 'MADRAS NOTEBOOK MEDIUM', 'SE
T OF 3 WOODEN HEART DECORATIONS', 'FAMILY ALBUM WHITE PICTURE FRAME', 'REX CASH+CARRY JUMBO SHOPPER'], ['COFFEE MUG PEARS  DESI
GN', 'TRAVEL CARD WALLET VINTAGE TICKET', 'AIRLINE BAG VINTAGE JET SET RED', 'AIRLINE BAG VINTAGE JET SET WHITE', 'GREY HEART H
OT WATER BOTTLE', 'LOVE HOT WATER BOTTLE', 'TRAVEL CARD WALLET I LOVE LONDON', 'KNITTED UNION FLAG HOT WATER BOTTLE', 'HOT WATE
R BOTTLE I AM SO POORLY', 'AIRLINE BAG VINTAGE WORLD CHAMPION ', 'AIRLINE BAG VINTAGE TOKYO 78', 'HOT WATER BOTTLE TEA AND SYMP
ATHY', 'BLUE PAISLEY POCKET BOOK', 'ABSTRACT CIRCLES POCKET BOOK', 'HAND WARMER RED RETROSPOT', 'PLASTERS IN TIN WOODLAND ANIMA
LS', 'PLASTERS IN TIN VINTAGE PAISLEY ', 'HAND WARMER SCOTTY DOG DESIGN', 'HAND WARMER BIRD DESIGN', 'PLASTERS IN TIN STRONGMA
N', 'PLASTERS IN TIN CIRCUS PARADE ']]
```

Figure 8.15: Four elements of the list of lists, where each sub-list contains all the items belonging to an individual invoice

```
[[False False False ... False False False]
 [False False False ... False False False]
 [False False False ... False False False]
 ...
 [False False False ... False False False]
 [False False False ... False False False]
 [False False False ... False False False]]
```

Figure 8.16: The multi-dimensional array containing the Boolean variables indicating product presence in each transaction

	4 PURPLE FLOCK DINNER CANDLES	50'S CHRISTMAS GIFT BAG LARGE	DOLLY GIRL BEAKER	I LOVE LONDON MINI BACKPACK	NINE DRAWER OFFICE TIDY	OVAL WALL MIRROR DIAMANTE	RED SPOT GIFT BAG LARGE	SET 2 TEA TOWELS I LOVE LONDON
4970	False	False	False	False	False	False	False	False
4971	False	False	True	False	False	False	False	False
4972	False	False	False	False	False	False	False	False
4973	False	False	False	False	False	False	False	False
4974	False	False	False	False	False	False	False	False
4975	False	False	False	False	False	False	False	False
4976	False	False	False	False	False	False	False	False
4977	False	False	False	False	False	False	False	False
4978	False	False	False	False	False	False	False	False
4979	False	False	False	False	False	False	False	False

Figure 8.17: A small section of the encoded data recast as a DataFrame

Figure 8.18: A subset of the cleaned, encoded, and recast DataFrame built from the complete online retail dataset

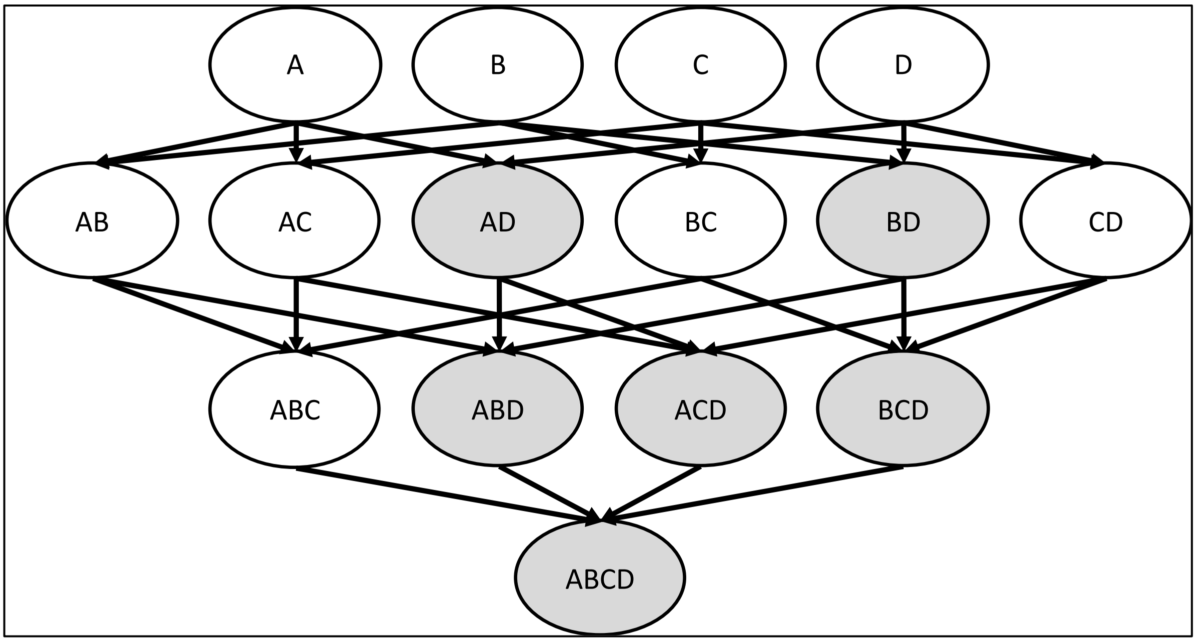


Figure 8.19: A mapping of how item sets are built and how the Apriori principle can greatly decrease the computational requirements (all the grayed-out nodes are infrequent)

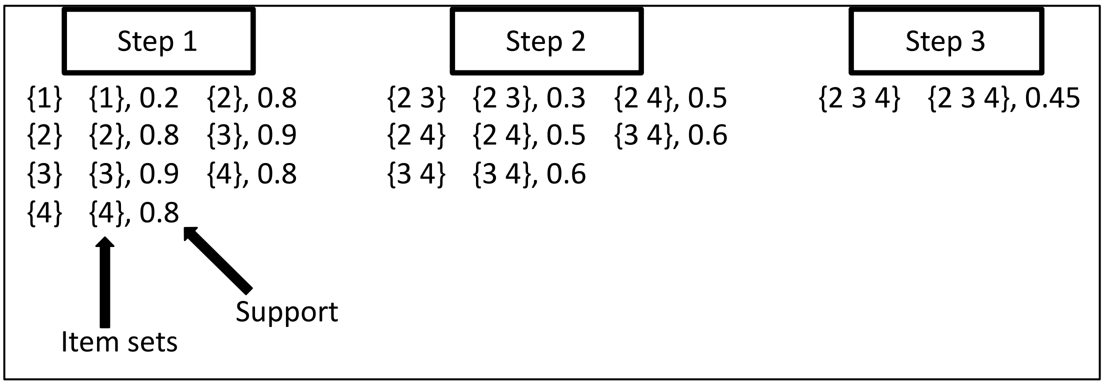


Figure 8.20: Assuming a minimum support threshold of 0.4, the diagram shows the general Apriori algorithm structure

	support	itemsets
0	0.0168	(2)
1	0.0150	(10)
2	0.0116	(15)
3	0.0144	(18)
4	0.0210	(19)
5	0.0144	(20)
6	0.0138	(21)

Figure 8.21: Basic output of the Apriori algorithm run using mlxtend

	support	itemsets
0	0.0168	(DOLLY GIRL BEAKER)
1	0.0150	(10 COLOUR SPACEBOY PEN)
2	0.0116	(12 MESSAGE CARDS WITH ENVELOPES)
3	0.0144	(12 PENCILS SMALL TUBE SKULL)
4	0.0210	(12 PENCILS TALL TUBE POSY)
5	0.0144	(12 PENCILS TALL TUBE RED RETROSPOT)
6	0.0138	(12 PENCILS TALL TUBE SKULLS)

Figure 8.22: The output of the Apriori algorithm with the actual item names instead of numerical designations

	support	itemsets	length
0	0.0168	(DOLLY GIRL BEAKER)	1
1	0.0150	(10 COLOUR SPACEBOY PEN)	1
2	0.0116	(12 MESSAGE CARDS WITH ENVELOPES)	1
3	0.0144	(12 PENCILS SMALL TUBE SKULL)	1
4	0.0210	(12 PENCILS TALL TUBE POSY)	1
5	0.0144	(12 PENCILS TALL TUBE RED RETROSPOT)	1
6	0.0138	(12 PENCILS TALL TUBE SKULLS)	1

Figure 8.23: The Apriori algorithm output plus an additional column containing the lengths of the item sets

	support	itemsets	length
1	0.015	(10 COLOUR SPACEBOY PEN)	1

Figure 8.24: The output DataFrame filtered down to a single item set

	support	itemsets	length
837	0.0202	(ALARM CLOCK BAKELIKE IVORY, ALARM CLOCK BAKEL...	2
956	0.0202	(LUNCH BAG APPLE DESIGN, CHARLOTTE BAG APPLES ...	2
994	0.0200	(LUNCH BAG PINK POLKADOT, CHARLOTTE BAG PINK P...	2
1026	0.0206	(CHARLOTTE BAG SUKI DESIGN, LUNCH BAG BLACK S...	2
1032	0.0206	(CHARLOTTE BAG SUKI DESIGN, LUNCH BAG RED RETR...	2
1131	0.0200	(JUMBO SHOPPER VINTAGE RED PAISLEY, DOTCOM POS...	2
1298	0.0208	(HEART OF WICKER LARGE, HEART OF WICKER SMALL)	2
1305	0.0200	(HEART OF WICKER SMALL, SMALL WHITE HEART OF W...	2
1316	0.0204	(JAM MAKING SET PRINTED, JAM MAKING SET WITH J...	2
1349	0.0208	(SET OF 3 REGENCY CAKE TINS, JAM MAKING SET PR...	2
1440	0.0200	(JUMBO BAG ALPHABET, LUNCH BAG ALPHABET DESIGN)	2
1464	0.0206	(JUMBO BAG APPLES, JUMBO BAG DOILEY PATTERNS)	2
1471	0.0202	(JUMBO BAG SCANDINAVIAN BLUE PAISLEY, JUMBO BA...	2
1472	0.0202	(JUMBO BAG SPACEBOY DESIGN, JUMBO BAG APPLES)	2
1479	0.0204	(JUMBO BAG APPLES, JUMBO STORAGE BAG SKULLS)	2
1575	0.0200	(JUMBO BAG PINK POLKADOT, JUMBO BAG OWLS)	2
1583	0.0208	(JUMBO BAG WOODLAND ANIMALS, JUMBO BAG OWLS)	2

Figure 8.25: The Apriori algorithm output DataFrame filtered by length and support

```
Text(0.5, 1.0, 'Support')
```

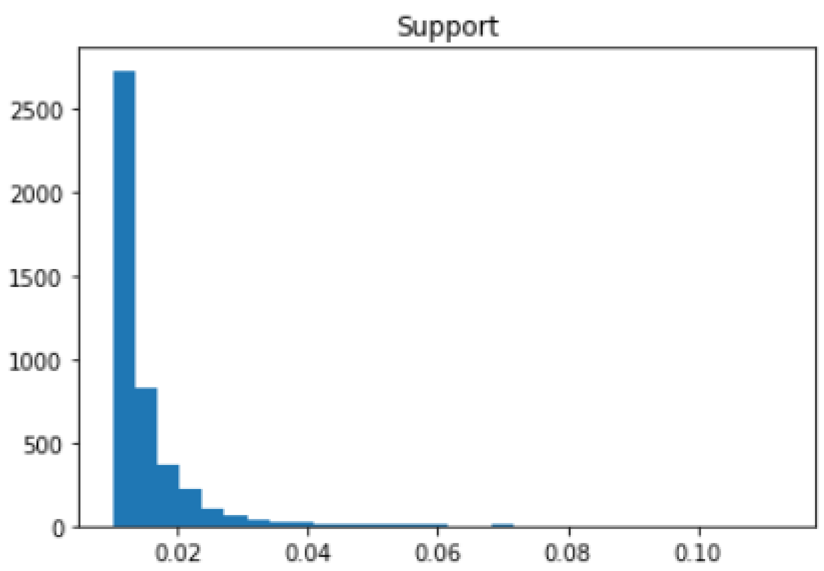


Figure 8.26: Distribution of the support values returned by the Apriori algorithm

```
Text(0.5, 1.0, 'Support')
```

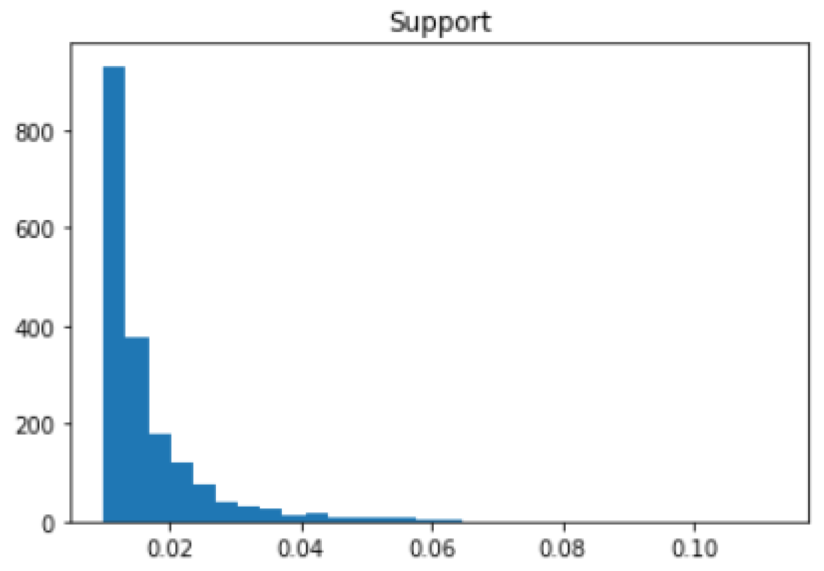


Figure 8.27: Distribution of support values

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(DOLLY GIRL BEAKER)	(SPACEBOY BEAKER)	0.0168	0.0172	0.0126	0.750000	43.604651	0.012311	3.931200
1	(SPACEBOY BEAKER)	(DOLLY GIRL BEAKER)	0.0172	0.0168	0.0126	0.732558	43.604651	0.012311	3.676313
2	(ALARM CLOCK BAKELIKE CHOCOLATE)	(ALARM CLOCK BAKELIKE GREEN)	0.0208	0.0580	0.0160	0.769231	13.262599	0.014794	4.082000
3	(ALARM CLOCK BAKELIKE CHOCOLATE)	(ALARM CLOCK BAKELIKE RED)	0.0208	0.0498	0.0142	0.682692	13.708681	0.013164	2.994570
4	(ALARM CLOCK BAKELIKE IVORY)	(ALARM CLOCK BAKELIKE GREEN)	0.0302	0.0580	0.0202	0.668874	11.532313	0.018448	2.844840
5	(ALARM CLOCK BAKELIKE ORANGE)	(ALARM CLOCK BAKELIKE GREEN)	0.0282	0.0580	0.0212	0.751773	12.961604	0.019564	3.794914
6	(ALARM CLOCK BAKELIKE PINK)	(ALARM CLOCK BAKELIKE GREEN)	0.0380	0.0580	0.0254	0.668421	11.524501	0.023196	2.840952

Figure 8.28: The first 7 rows of the association rules generated using only 5,000 transactions

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(POPPY'S PLAYHOUSE KITCHEN, POPPY'S PLAYHOUSE ...	(POPPY'S PLAYHOUSE LIVINGROOM)	0.0136	0.0148	0.0102	0.750000	50.675676	0.009999	3.940800
1	(POPPY'S PLAYHOUSE LIVINGROOM)	(POPPY'S PLAYHOUSE KITCHEN, POPPY'S PLAYHOUSE ...	0.0148	0.0136	0.0102	0.689189	50.675676	0.009999	3.173635
2	(DOLLY GIRL CHILDRENS BOWL, SPACEBOY CHILDRENS...	(DOLLY GIRL CHILDRENS CUP, SPACEBOY CHILDRENS ...	0.0136	0.0140	0.0120	0.882353	63.025210	0.011810	8.381000
3	(DOLLY GIRL CHILDRENS CUP, SPACEBOY CHILDRENS ...	(DOLLY GIRL CHILDRENS BOWL, SPACEBOY CHILDRENS...	0.0140	0.0136	0.0120	0.857143	63.025210	0.011810	6.904800
4	(REGENCY TEA PLATE ROSES , GREEN REGENCY TEACU...	(REGENCY TEA PLATE GREEN , PINK REGENCY TEACUP...	0.0160	0.0138	0.0112	0.700000	50.724638	0.010979	3.287333
5	(REGENCY TEA PLATE GREEN , PINK REGENCY TEACUP...	(REGENCY TEA PLATE ROSES , GREEN REGENCY TEACU...	0.0138	0.0160	0.0112	0.811594	50.724638	0.010979	5.222769
6	(REGENCY TEA PLATE PINK, GREEN REGENCY TEACUP ...	(ROSES REGENCY TEACUP AND SAUCER , REGENCY TEA...	0.0124	0.0166	0.0106	0.854839	51.496308	0.010394	6.774533

Figure 8.29: The first 7 rows of the association rules

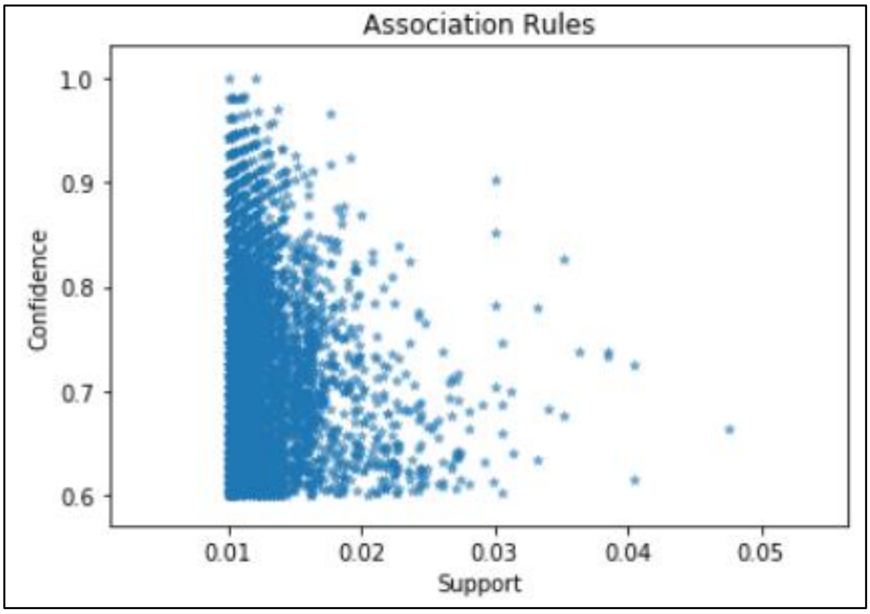


Figure 8.30: A plot of confidence against support

```
Text(0.5, 1.0, 'Confidence')
```

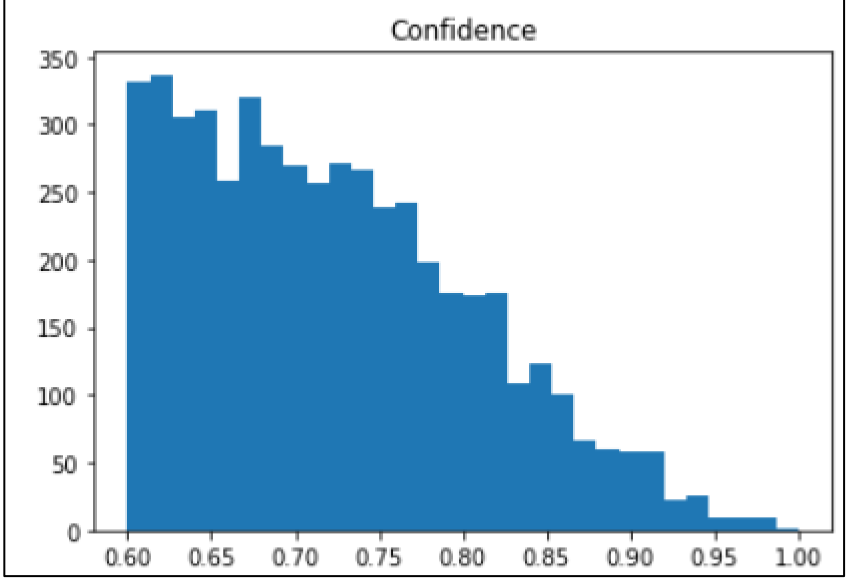


Figure 8.31: The distribution of confidence values

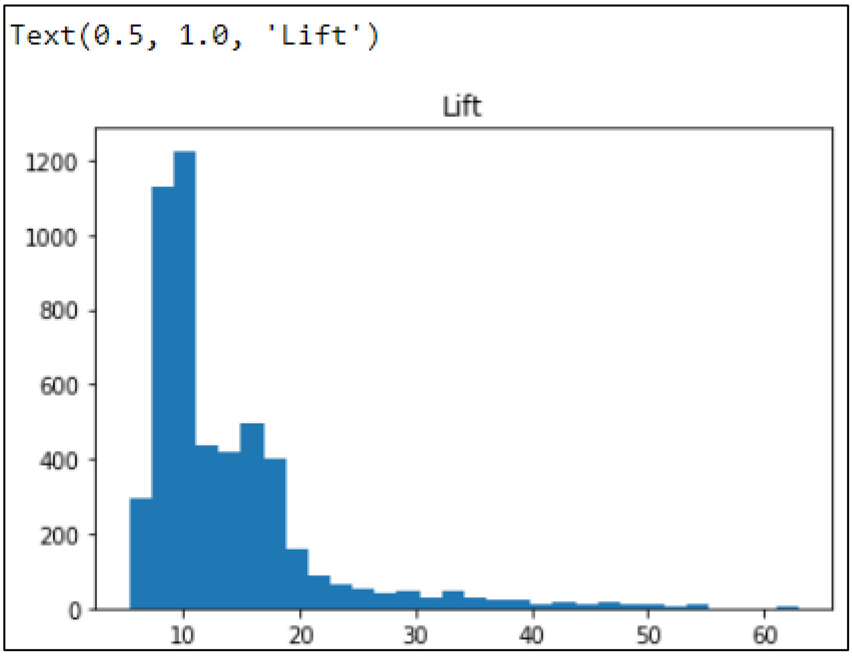


Figure 8.32: The distribution of lift values

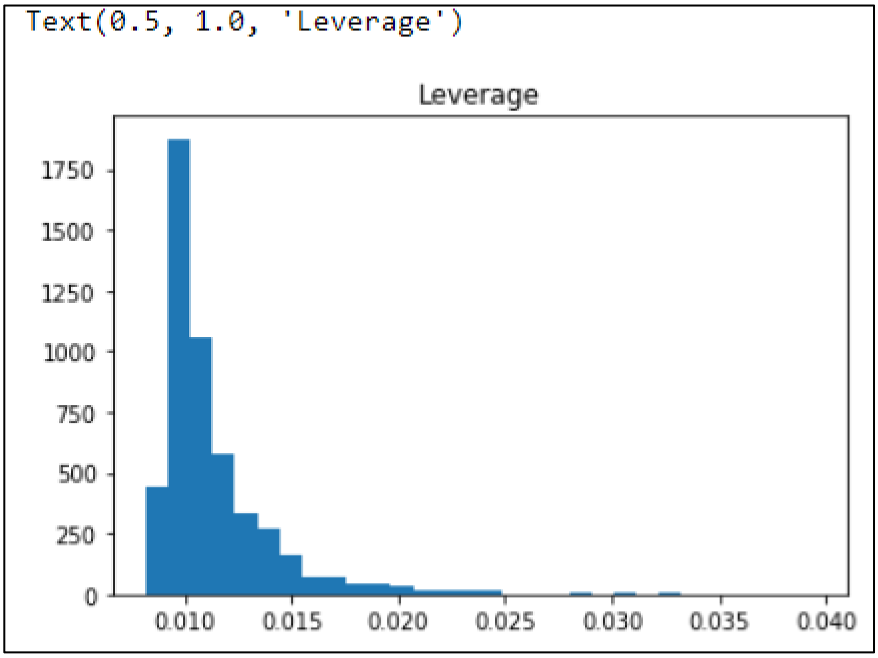


Figure 8.33: The distribution of leverage values

```
Text(0.5, 1.0, 'Conviction')
```

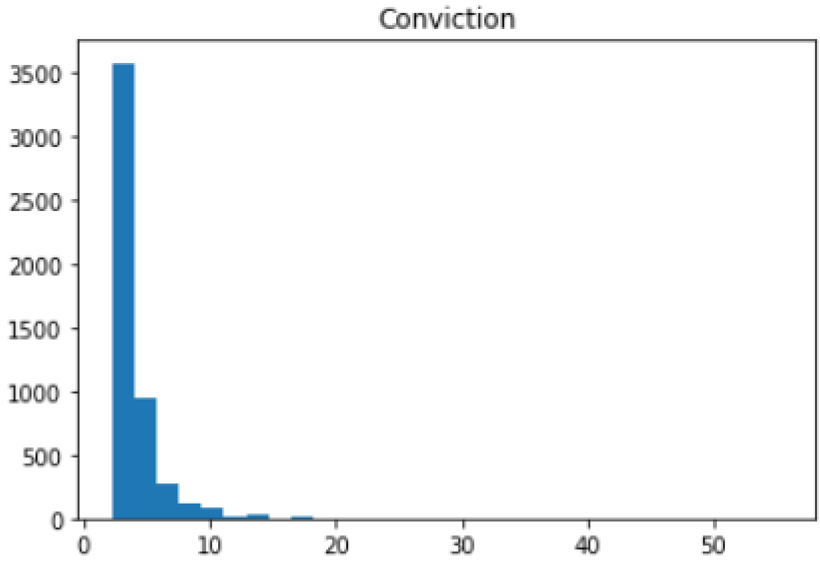


Figure 8.34: The distribution of conviction values

Lesson 9: Hotspot Analysis

Figure 9.1: A fabricated example of fire location data showing some potential hotspots

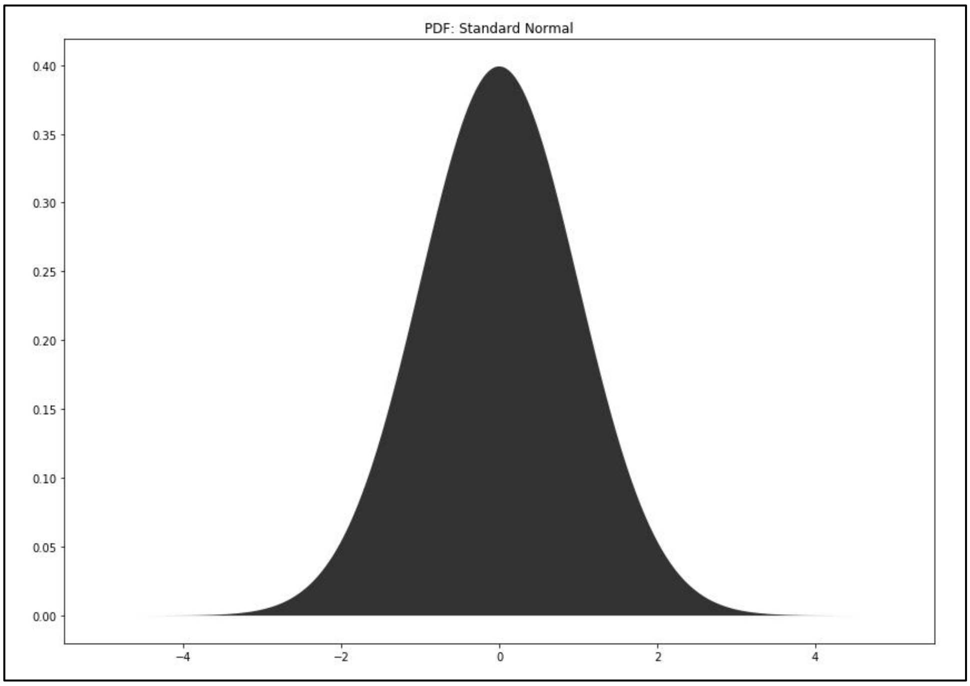


Figure 9.2: The standard normal distribution

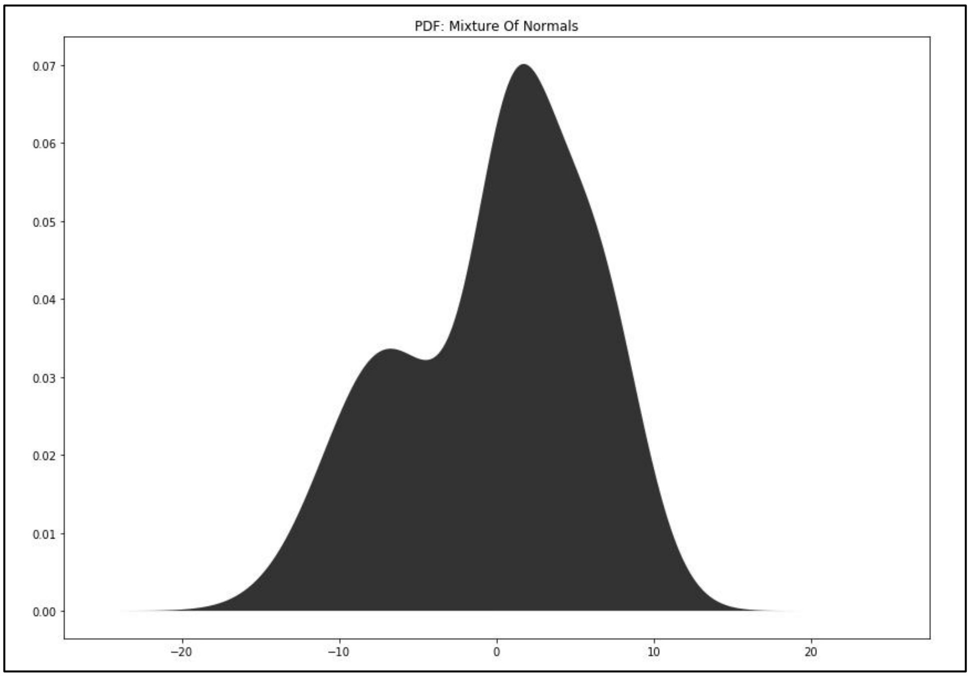


Figure 9.3: A mixture of three normal distributions

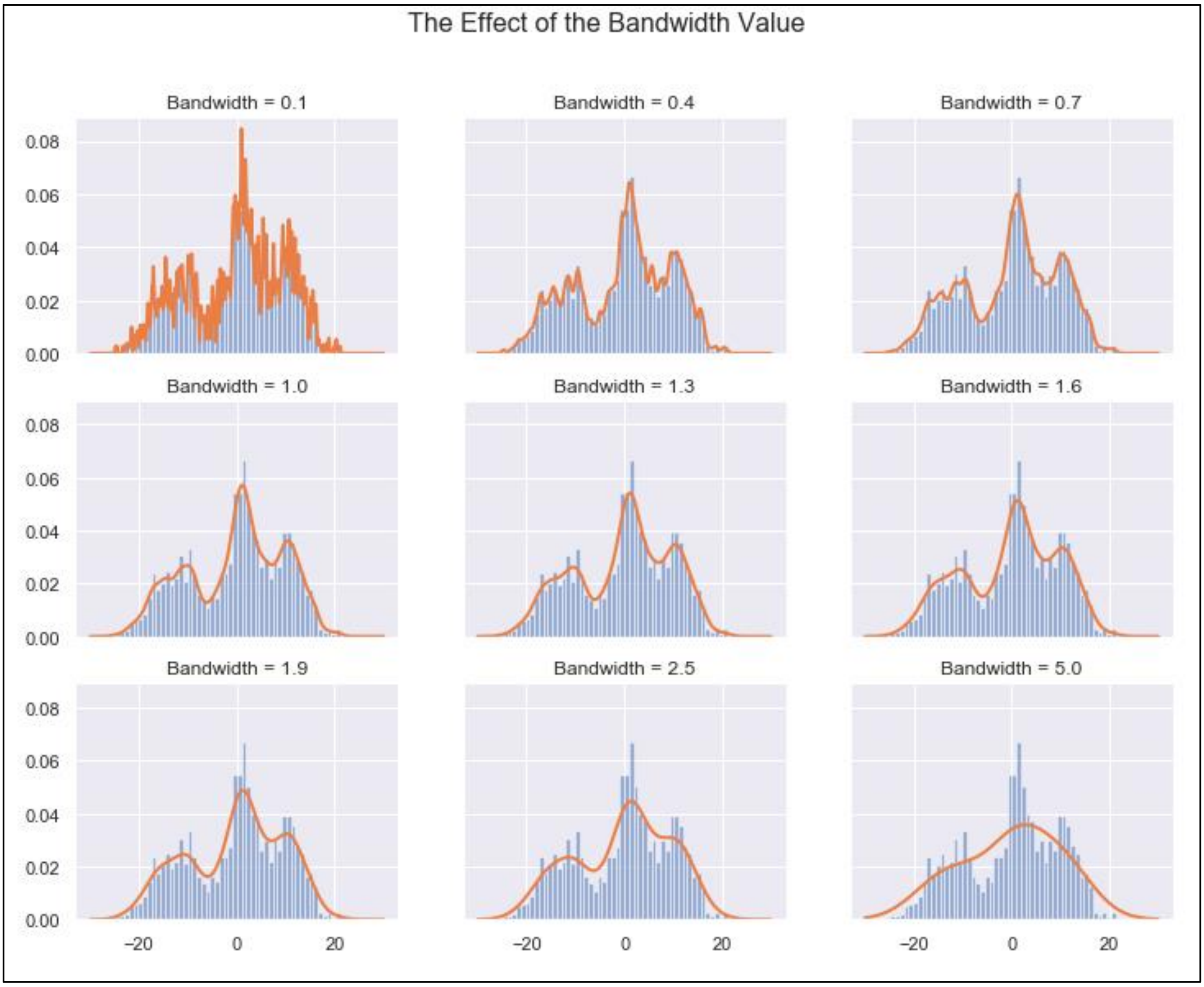


Figure 9.4: A 3 x 3 matrix of subplots; each of which features an estimated density created using one of nine bandwidth values

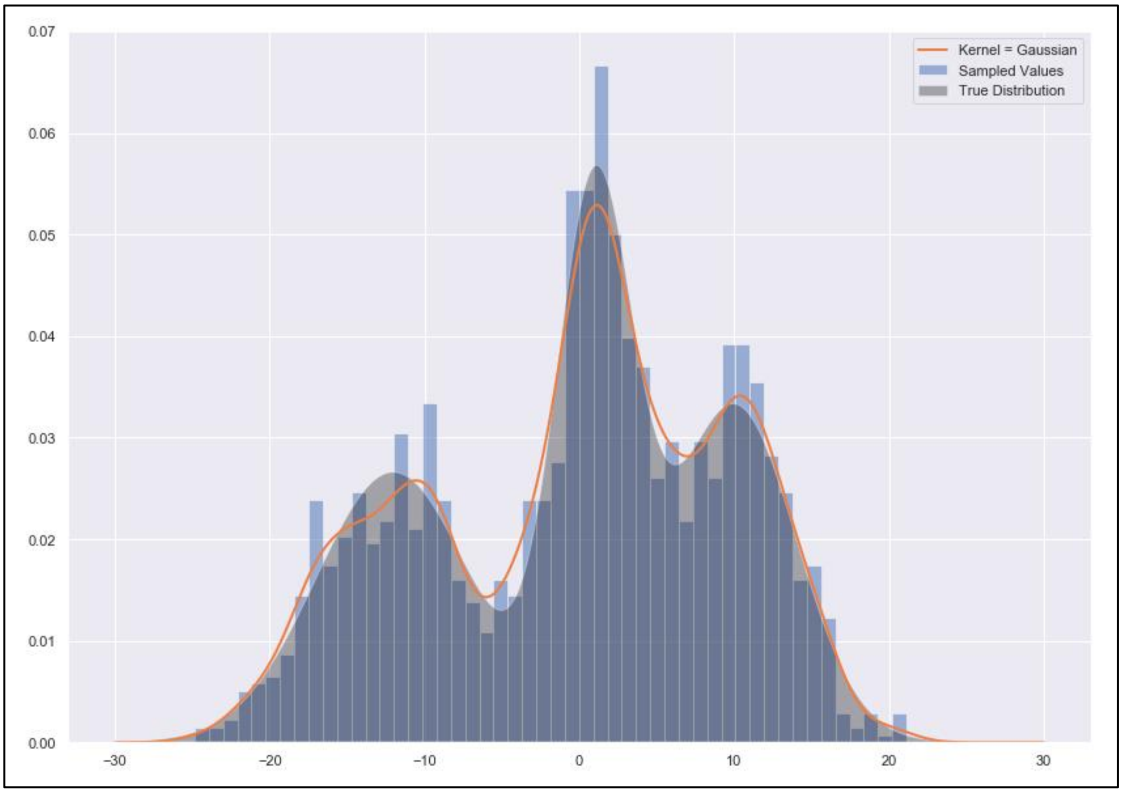


Figure 9.5: A histogram of the random sample with the true density and the optimal estimated density overlaid

$$K(x;h) \propto exp - \left(\frac{x^2}{2h^2}\right)$$

Figure 9.6: The formula for the Gaussian kernel

$$K(x;h) \propto \begin{Bmatrix} 0\ if\ |x| \geq h \\ 1\ if\ |x| < h \end{Bmatrix}$$

Figure 9.7: The formula for the Tophat kernel

$$K(x;h) \propto 1 - \frac{x^2}{h^2}$$

Figure 9.8: The formula for the Epanechnikov kernel

$$K(x;h) \propto exp\left(-\frac{|x|}{h}\right)$$

Figure 9.9: The formula for the Exponential kernel

$$K(x;h) \propto \begin{Bmatrix} 0\ if\ |x| \geq h \\ 1 - \frac{x}{h}\ if\ |x| < h \end{Bmatrix}$$

Figure 9.10: The formula for the Linear kernel

$$K(x;h) \propto \begin{Bmatrix} 0\ if\ |x| \geq h \\ cos\frac{\Pi x}{2h}\ if\ |x| < h \end{Bmatrix}$$

Figure 9.11: The formula for the Cosine kernel

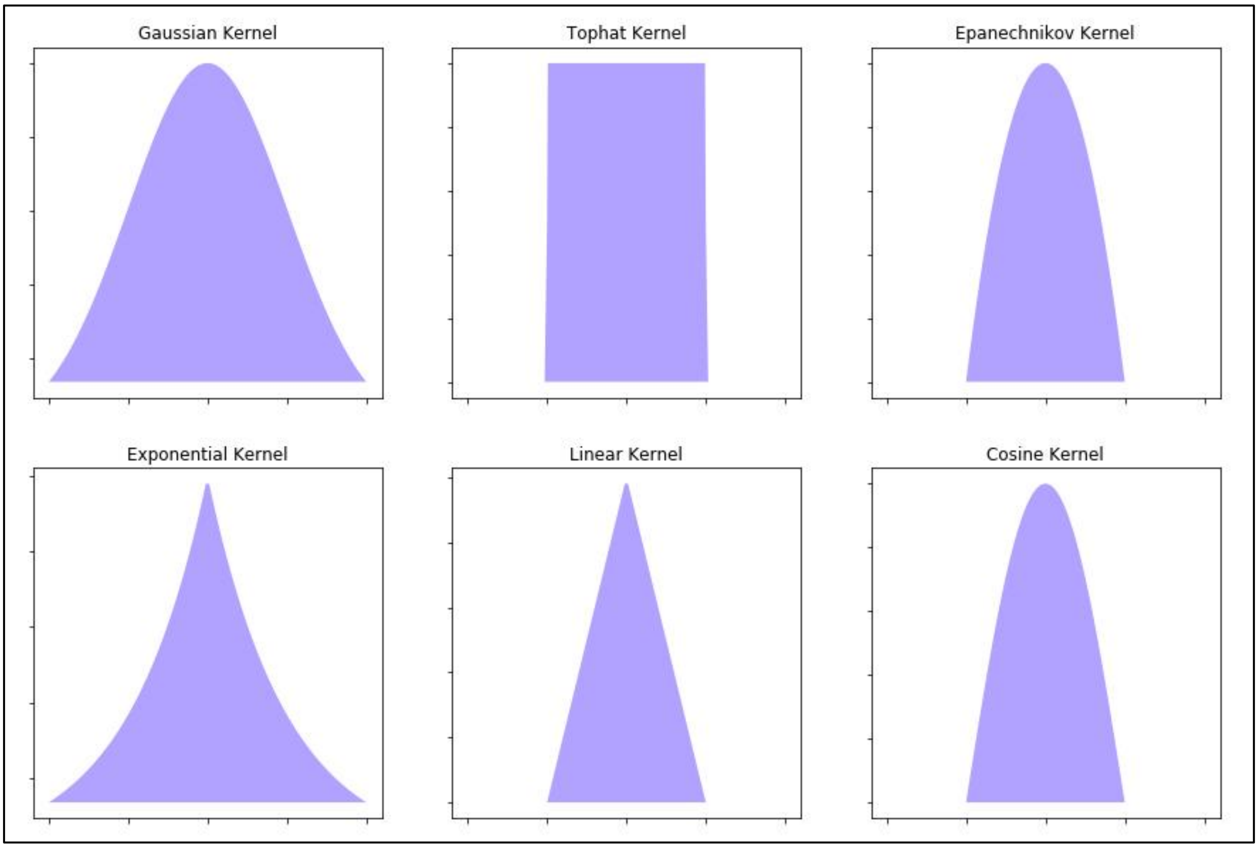


Figure 9.12: The general shapes of the six kernel functions

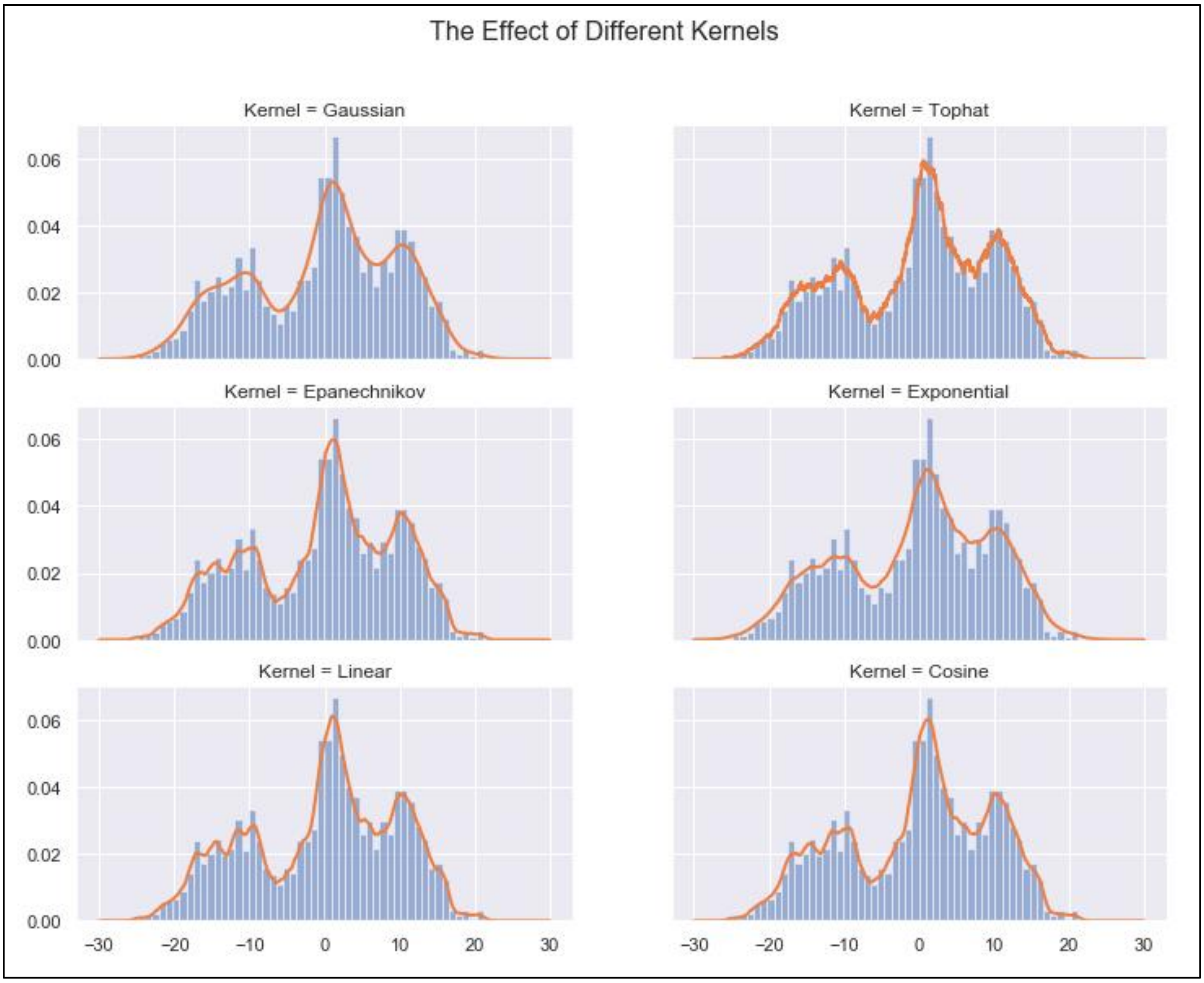


Figure 9.13: A 3 x 2 matrix of subplots, each of which features an estimated density created using one of six kernel functions

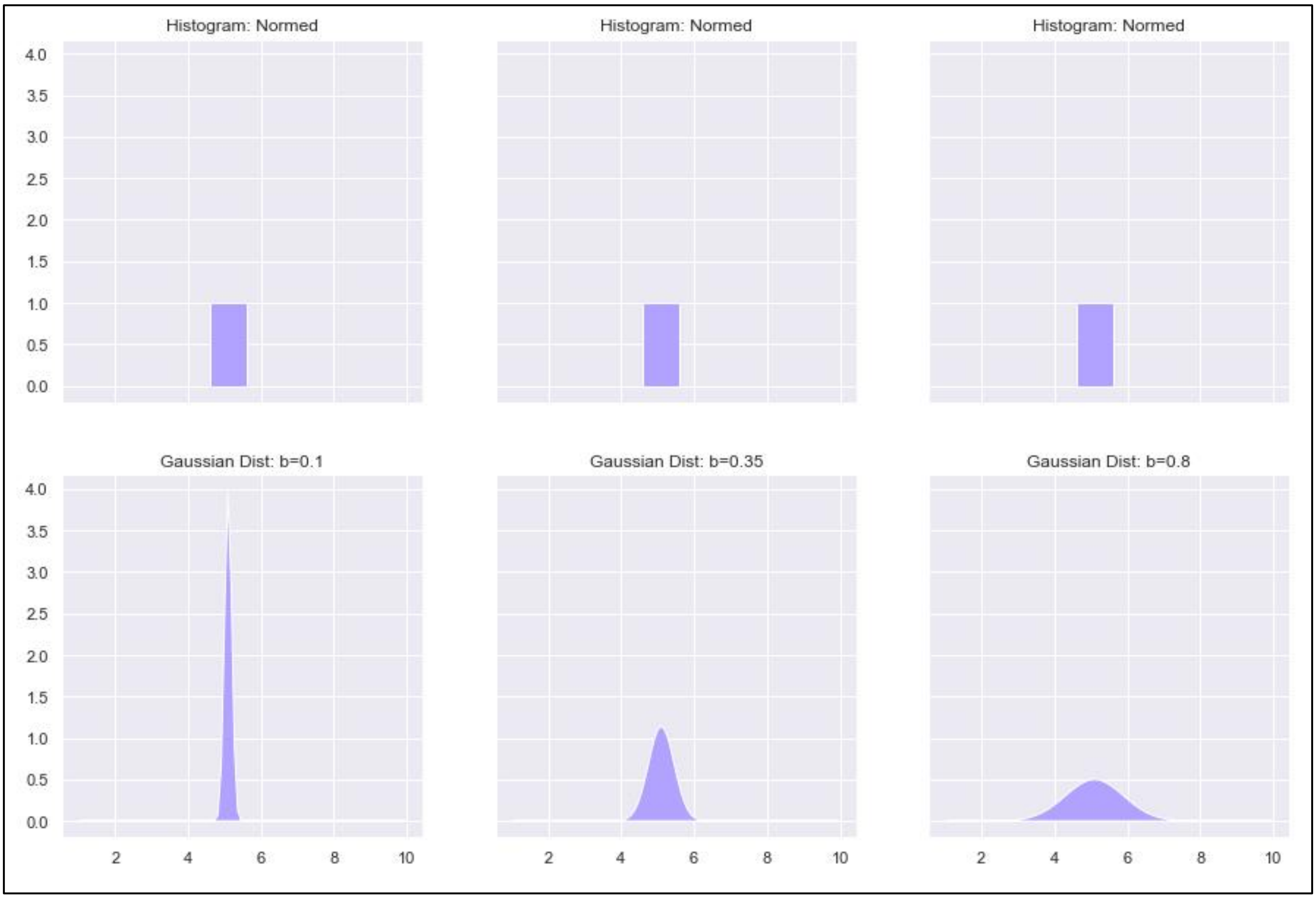


Figure 9.14: Showing one data point and its individual density at various bandwidth values

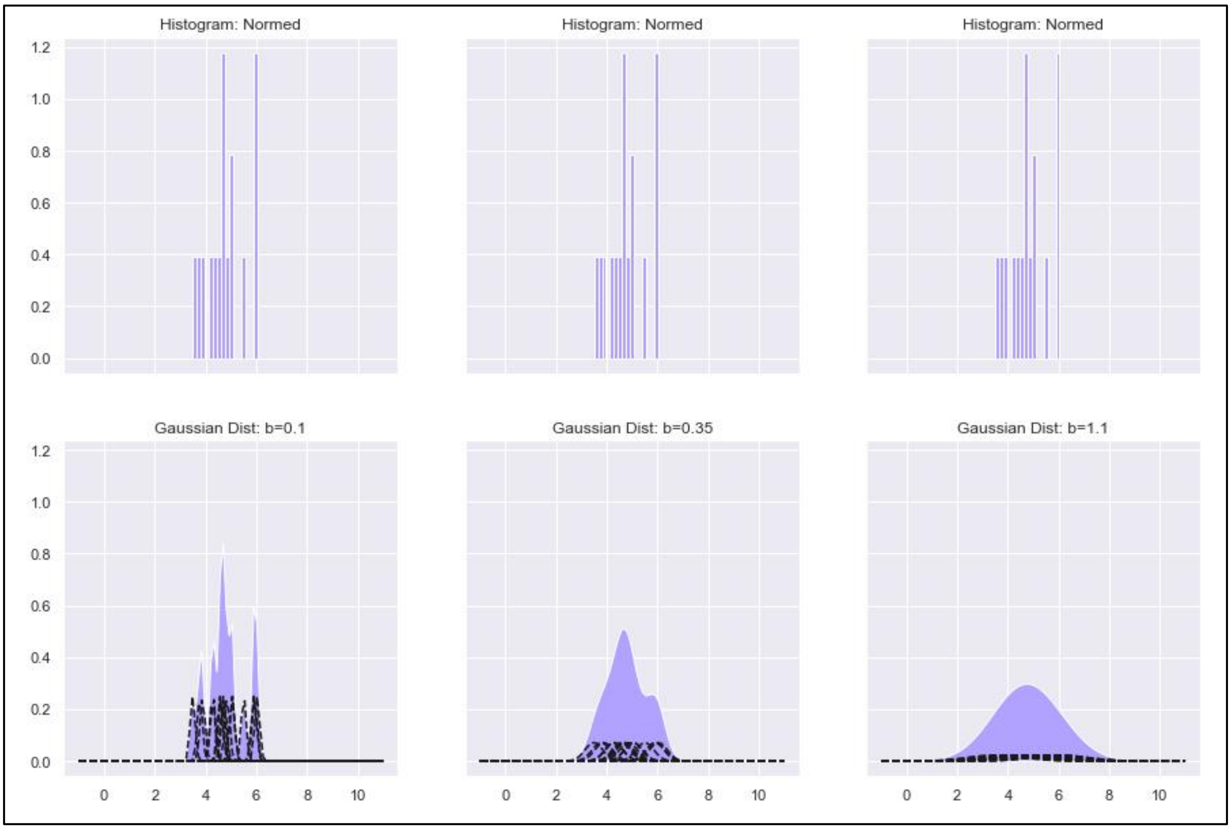


Figure 9.15: Showing 16 data points, their individual densities at various bandwidth values, and the sum of their individual densities

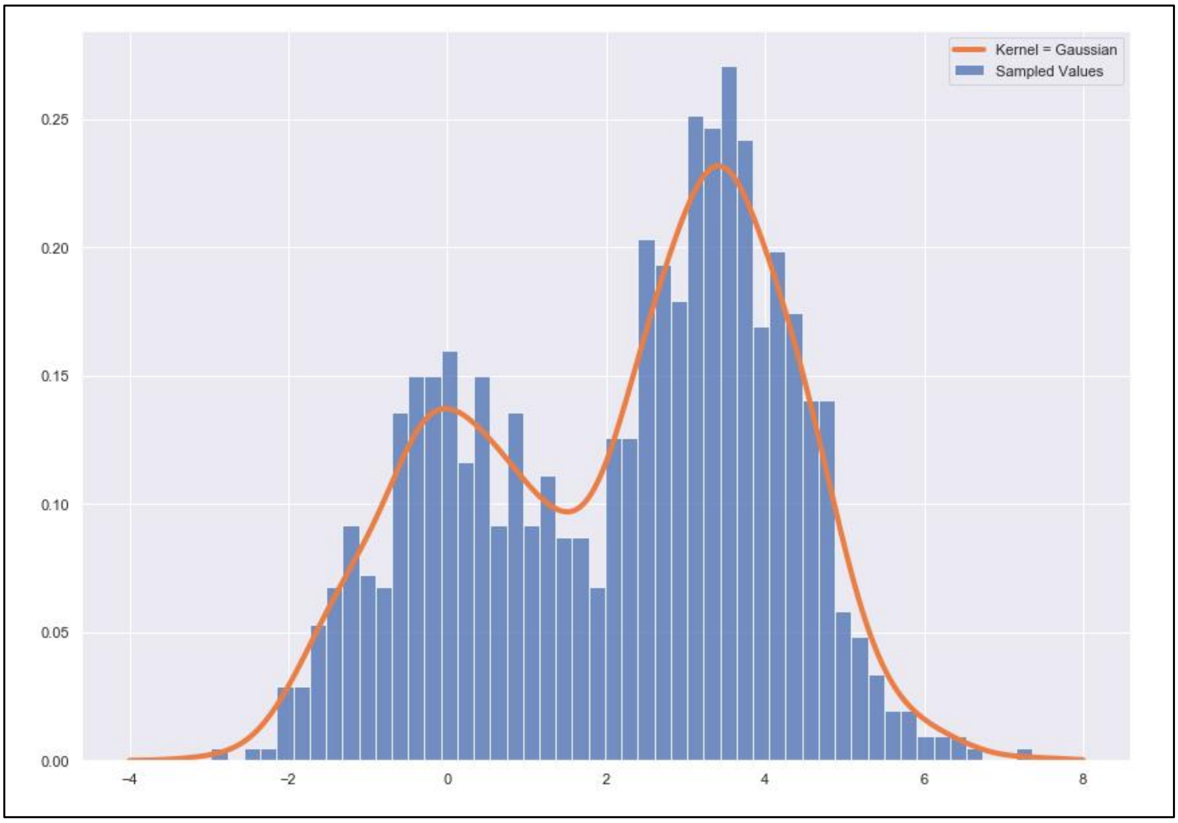


Figure 9.16: A histogram of the random sample with the optimal estimated density overlaid

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25

Figure 9.17: The first five rows of the California housing dataset from sklearn

	Latitude	Longitude
59	37.82	-122.29
87	37.81	-122.27
88	37.80	-122.27
391	37.90	-122.30
437	37.87	-122.30

Figure 9.18: The first five rows of the dataset filtered down to those rows that have a value of 15 or less in the HouseAge column

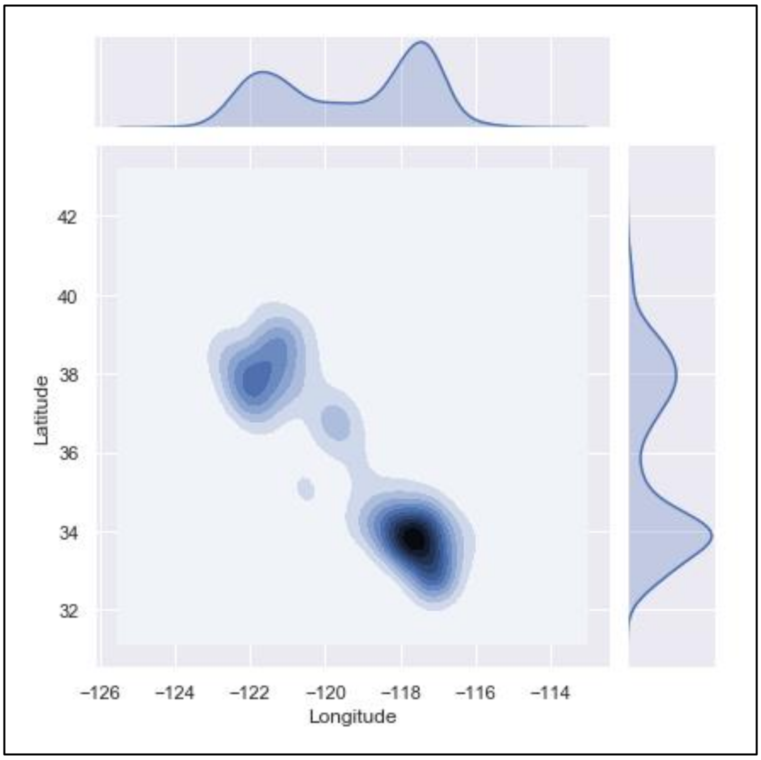


Figure 9.19: A joint plot containing both the two-dimensional estimated density plus the marginal densities for the dfLess15 dataset

	Latitude	Longitude
0	37.88	-122.23
2	37.85	-122.24
3	37.85	-122.25
4	37.85	-122.25
5	37.85	-122.25

Figure 9.20: The top of the dataset filtered to the rows containing values greater than 40 in the HouseAge column

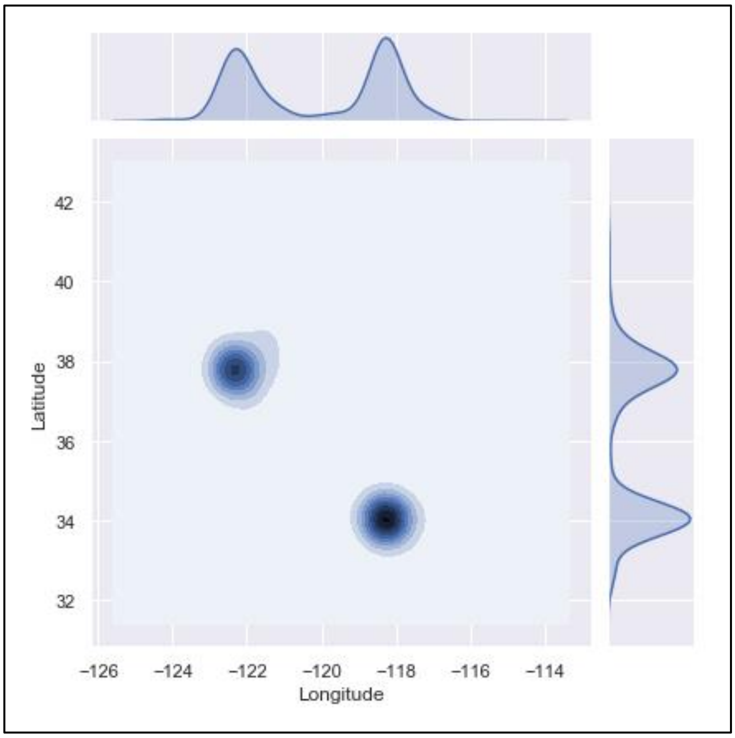


Figure 9.21: A joint plot containing both the two-dimensional estimated density plus the marginal densities for the dfMore40 dataset

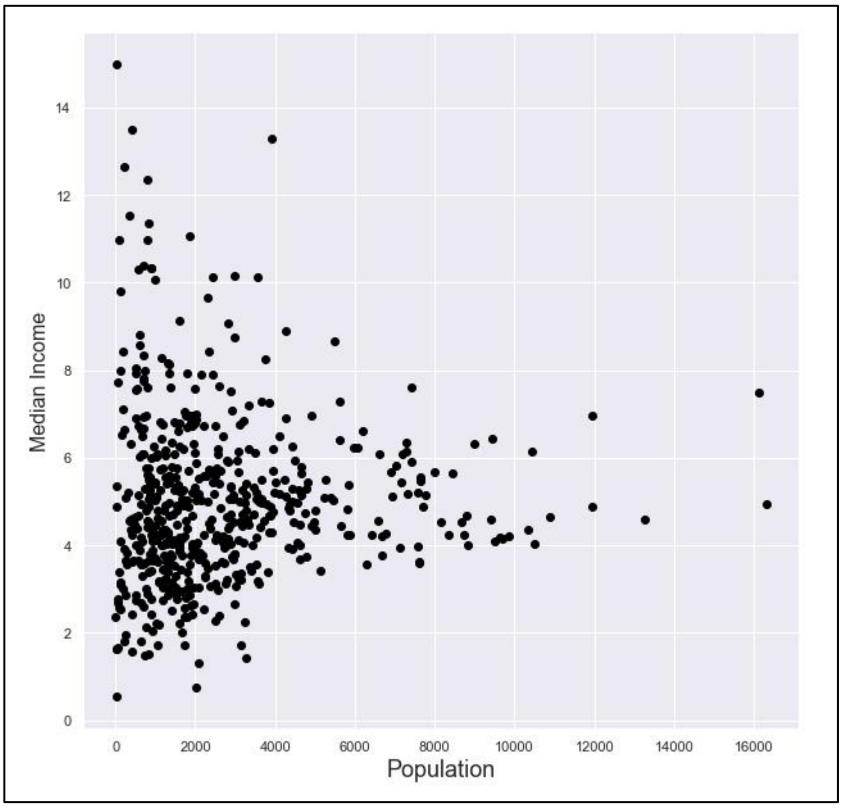


Figure 9.22: A scatterplot of the median income against population for values of five or less in the HouseAge column

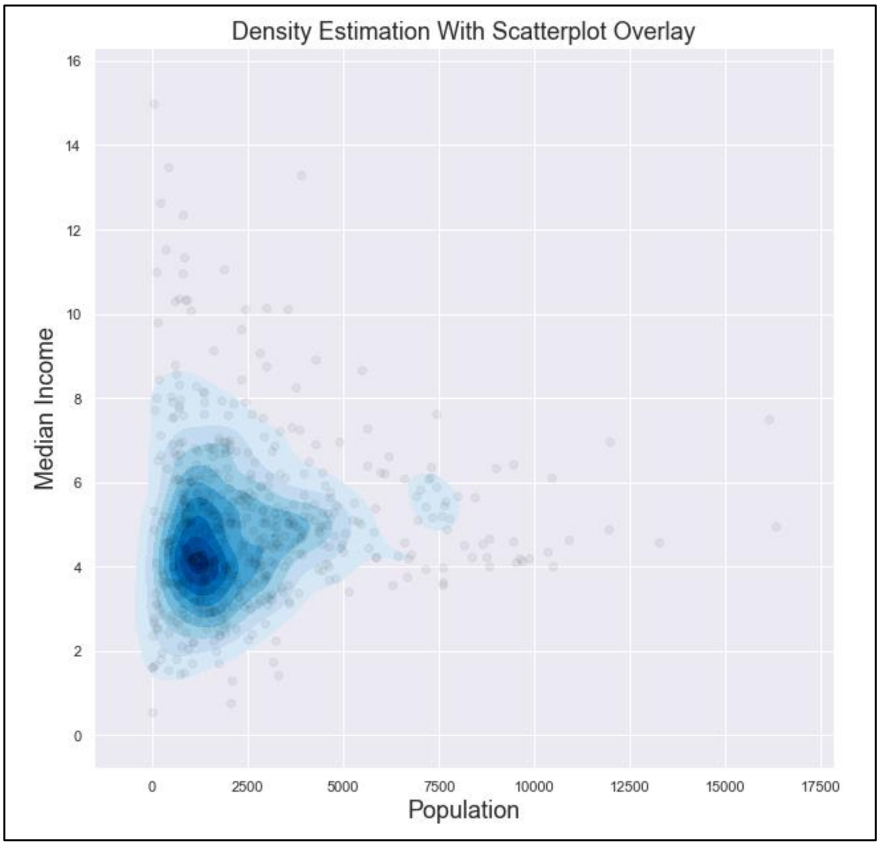


Figure 9.23: The same scatterplot as created in Step 6 with the estimated density overlaid

```
X Grid Component:
[[-124.23 -124.19 -124.17 ... -114.63 -114.57 -114.31]
 [-124.23 -124.19 -124.17 ... -114.63 -114.57 -114.31]
 [-124.23 -124.19 -124.17 ... -114.63 -114.57 -114.31]
 ...
 [-124.23 -124.19 -124.17 ... -114.63 -114.57 -114.31]
 [-124.23 -124.19 -124.17 ... -114.63 -114.57 -114.31]
 [-124.23 -124.19 -124.17 ... -114.63 -114.57 -114.31]]

Y Grid Component:
[[32.54 32.54 32.54 ... 32.54 32.54 32.54]
 [32.55 32.55 32.55 ... 32.55 32.55 32.55]
 [32.55 32.55 32.55 ... 32.55 32.55 32.55]
 ...
 [41.74 41.74 41.74 ... 41.74 41.74 41.74]
 [41.75 41.75 41.75 ... 41.75 41.75 41.75]
 [41.78 41.78 41.78 ... 41.78 41.78 41.78]]
```

Figure 9.24: The x and y components of the grid representing the dfLess15 dataset

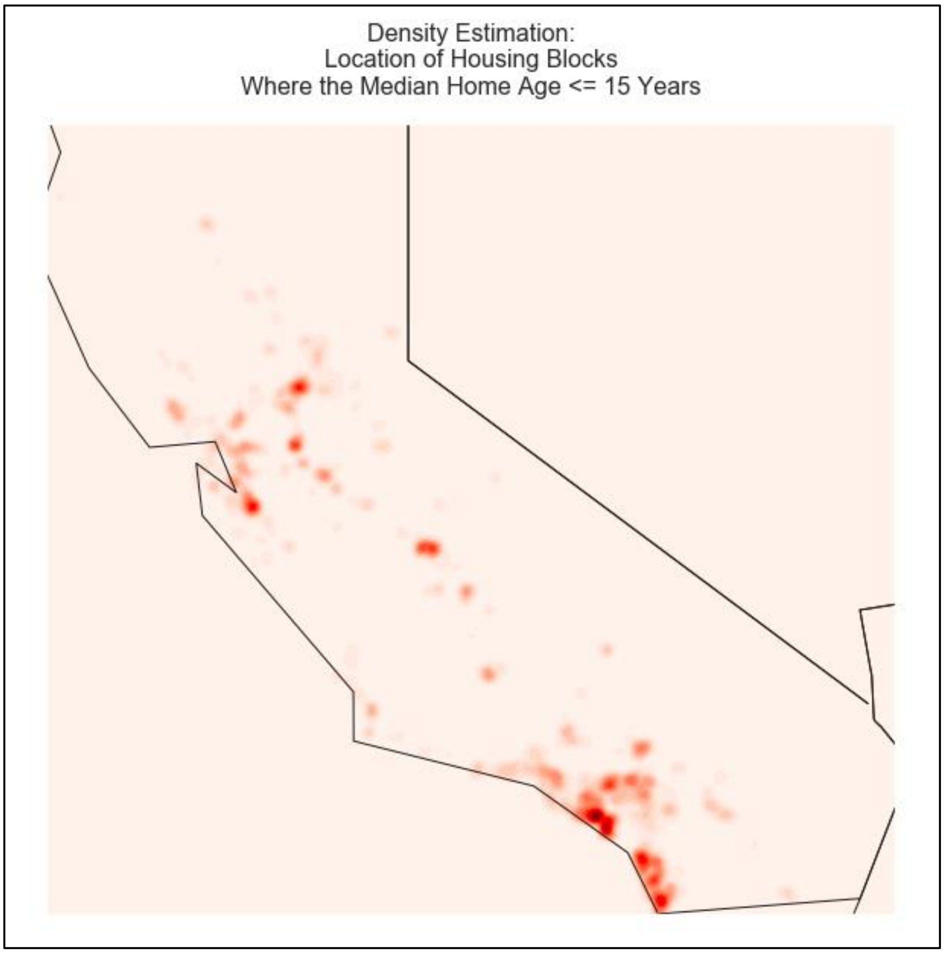


Figure 9.25: The estimated density of dfLess15 overlaid onto an outline of California

```
X Grid Component:
[[-124.35 -124.26 -124.23 ... -114.61 -114.6  -114.59]
 [-124.35 -124.26 -124.23 ... -114.61 -114.6  -114.59]
 [-124.35 -124.26 -124.23 ... -114.61 -114.6  -114.59]
 ...
 [-124.35 -124.26 -124.23 ... -114.61 -114.6  -114.59]
 [-124.35 -124.26 -124.23 ... -114.61 -114.6  -114.59]
 [-124.35 -124.26 -124.23 ... -114.61 -114.6  -114.59]]

Y Grid Component:
[[32.64 32.64 32.64 ... 32.64 32.64 32.64]
 [32.66 32.66 32.66 ... 32.66 32.66 32.66]
 [32.66 32.66 32.66 ... 32.66 32.66 32.66]
 ...
 [41.43 41.43 41.43 ... 41.43 41.43 41.43]
 [41.73 41.73 41.73 ... 41.73 41.73 41.73]
 [41.78 41.78 41.78 ... 41.78 41.78 41.78]]
```

Figure 9.26: The x and y components of the grid representing the dfMore40 dataset

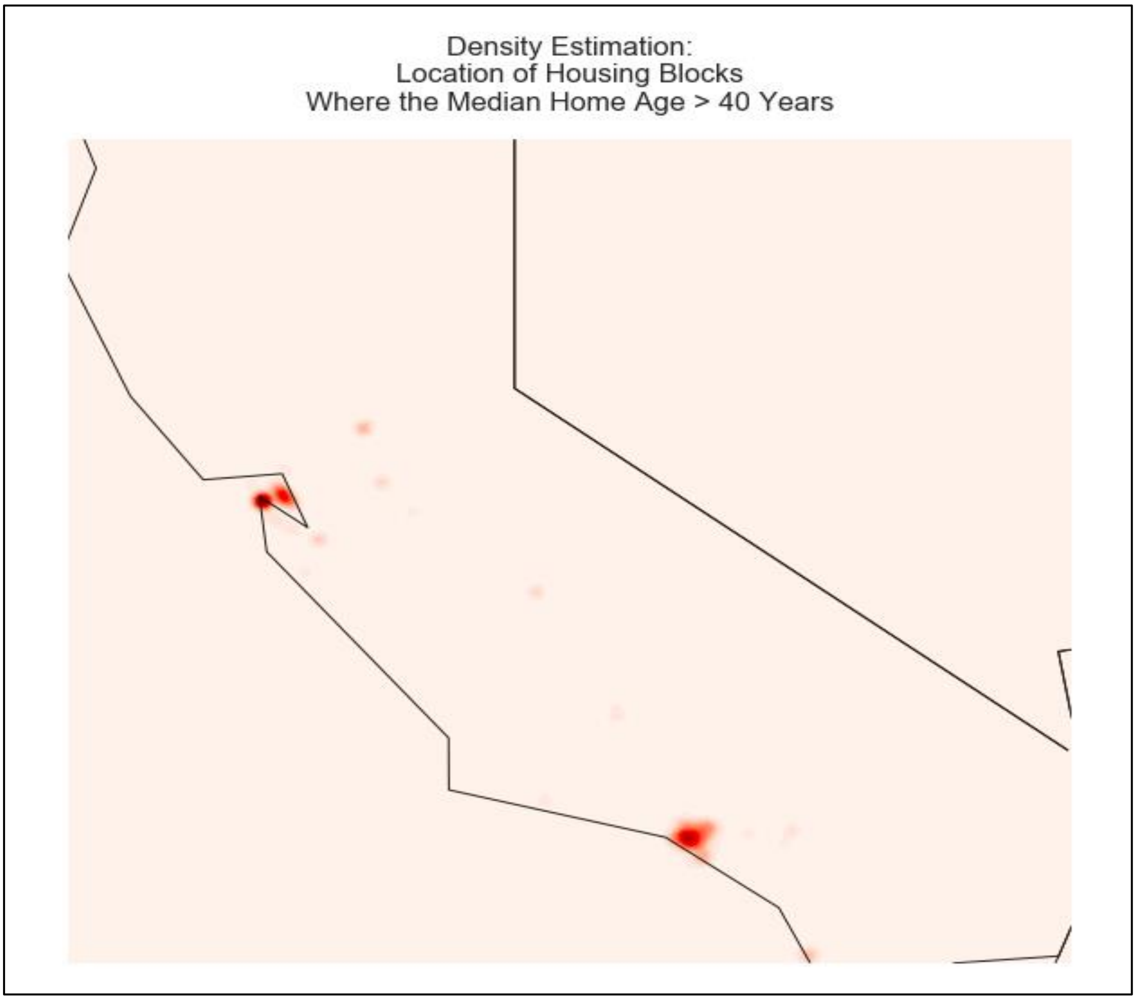


Figure 9.27: The estimated density of dfMore40 overlaid onto an outline of California

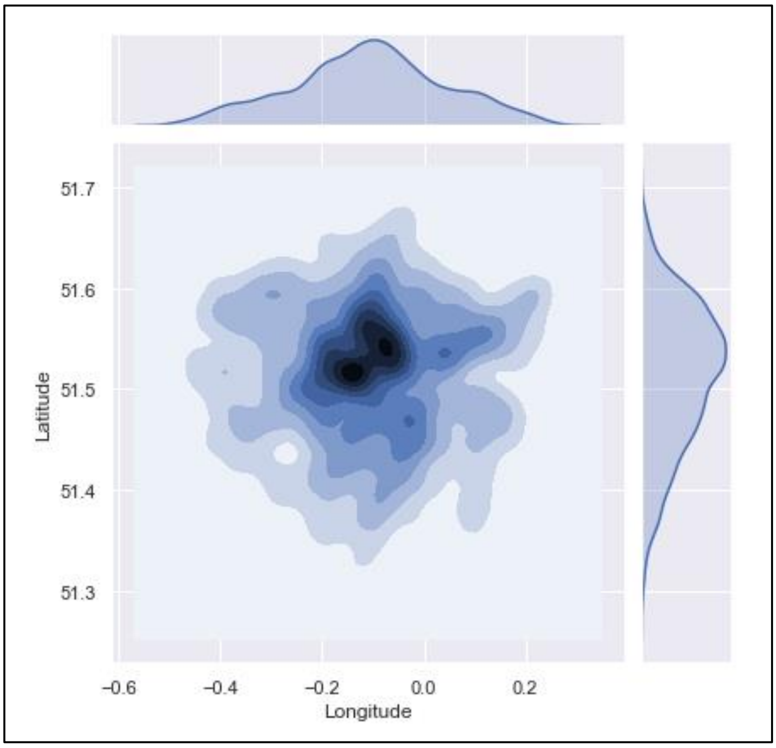


Figure 9.28: The estimated joint and marginal densities for burglaries in December 2018

Solutions

	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
0	5.1	3.5	1.4	0.2
1	4.9	3.0	1.4	0.2
2	4.7	3.2	1.3	0.2
3	4.6	3.1	1.5	0.2
4	5.0	3.6	1.4	0.2

Figure 1.22: First five rows of the data

```
[2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 0 1 0 0 0 0
 0 0 1 1 0 0 0 0 1 0 1 0 1 0 0 1 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 1 0 0 0 1 0
 0 1]
```

Figure 1.23: List of predicted species

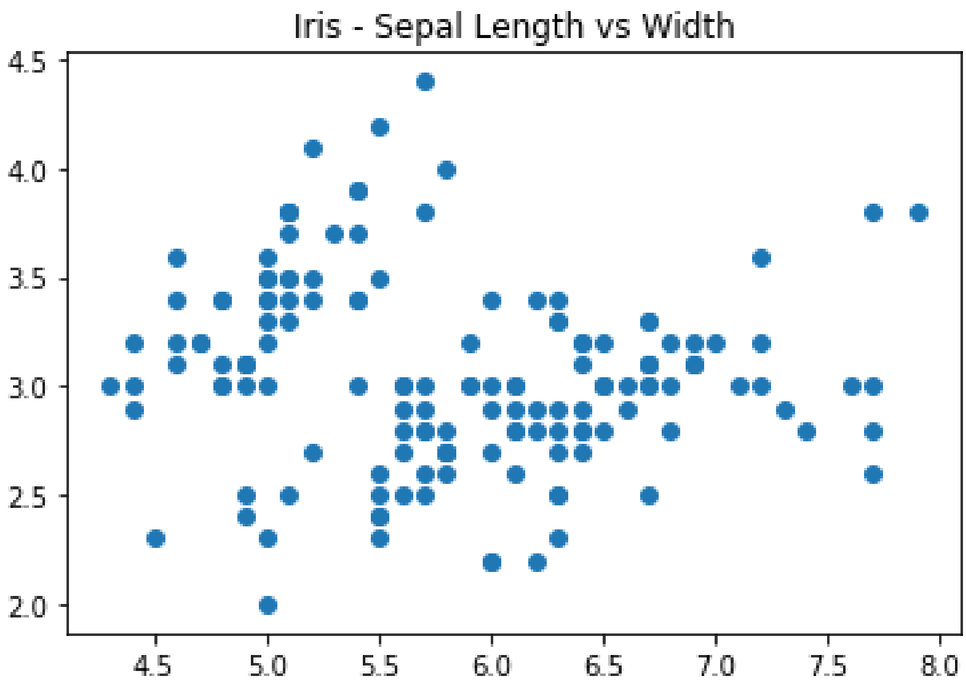


Figure 1.24: Plot of performed k-means implementation

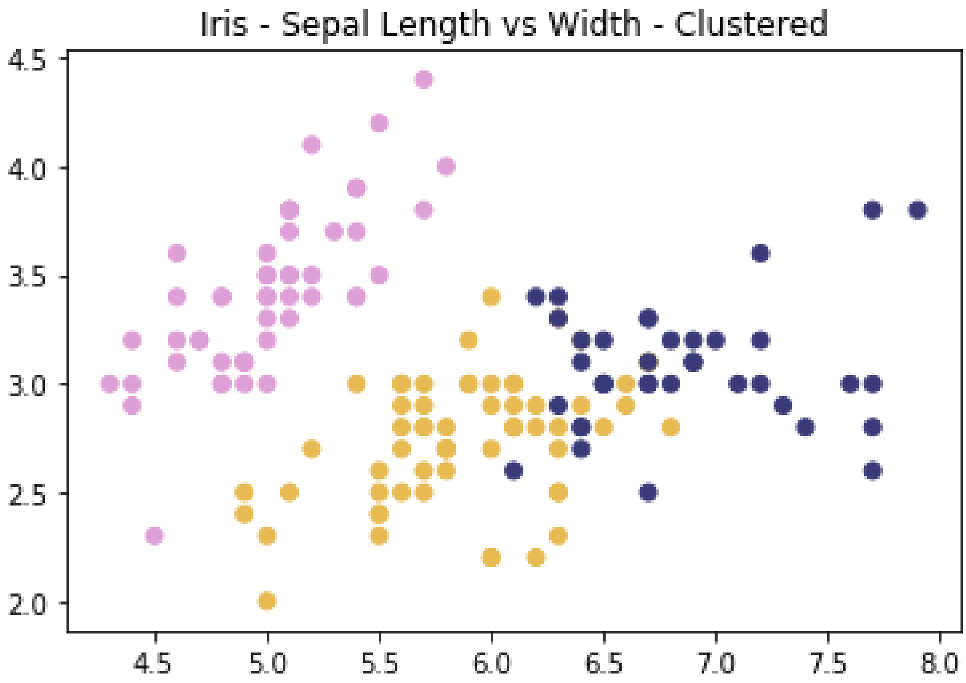


Figure 1.25: Clusters of Iris species

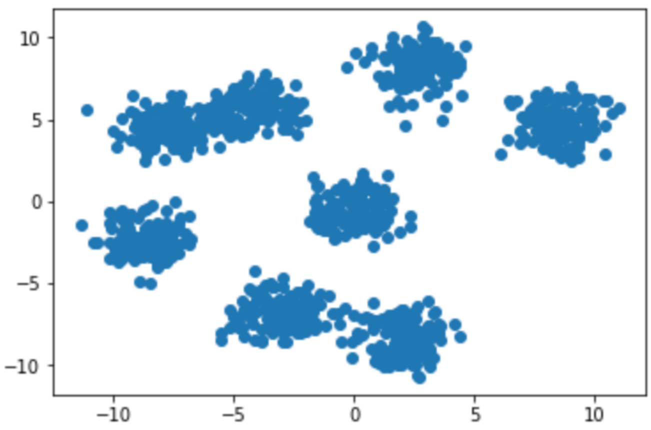


Figure 2.20: A scatter plot of the generated cluster dataset

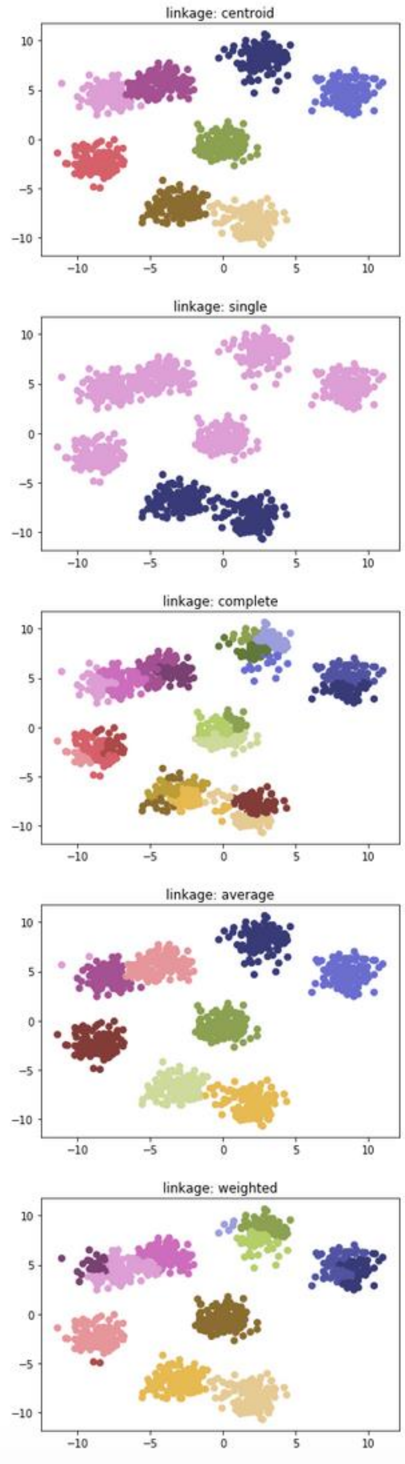


Figure 2.21: A scatter plot for all the methods

```
<bound method NDFrame.head of       OD_read  Proline
0         3.92   1065.0
1         3.40   1050.0
2         3.17   1185.0
3         3.45   1480.0
4         2.93    735.0
5         2.85   1450.0
```

Figure 2.22: The output of the wine dataset

Figure 2.23: A plot of raw wine data

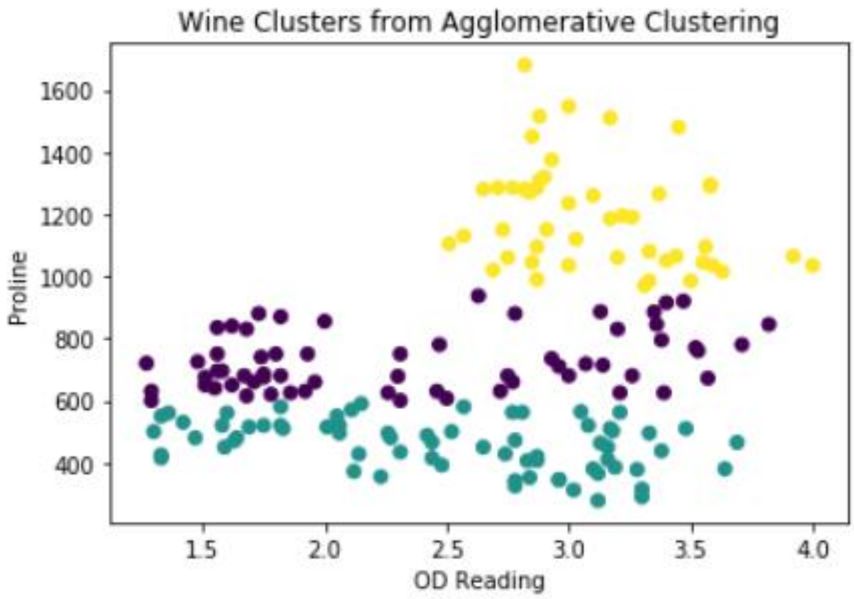


Figure 2.24: A plot of clusters from k-means clustering

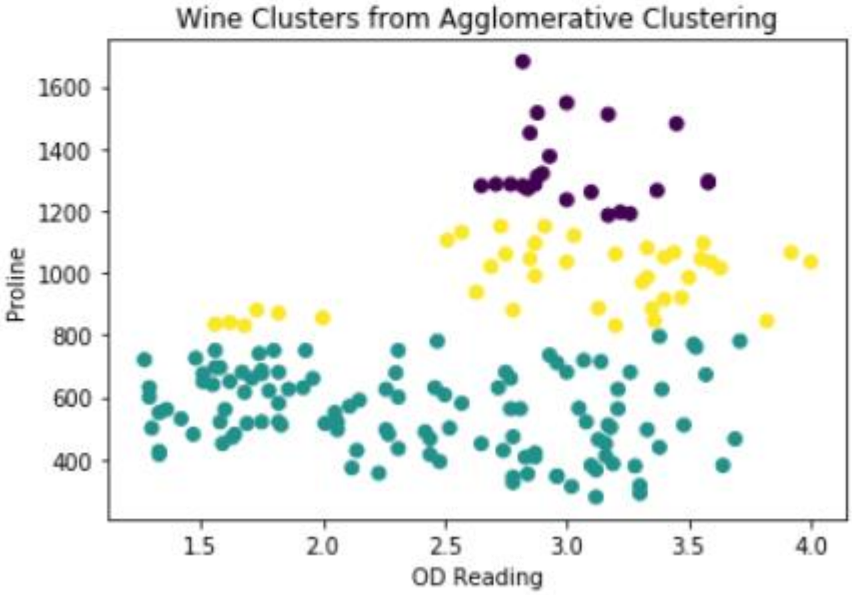


Figure 2.25: A plot of clusters from agglomerative clustering

```
Silhouette Scores for Wine Dataset:

K-Means Clustering:  0.5809421087616886
Agg Clustering:  0.5988495817462
```

Figure 2.26: Silhouette scores for the wine dataset

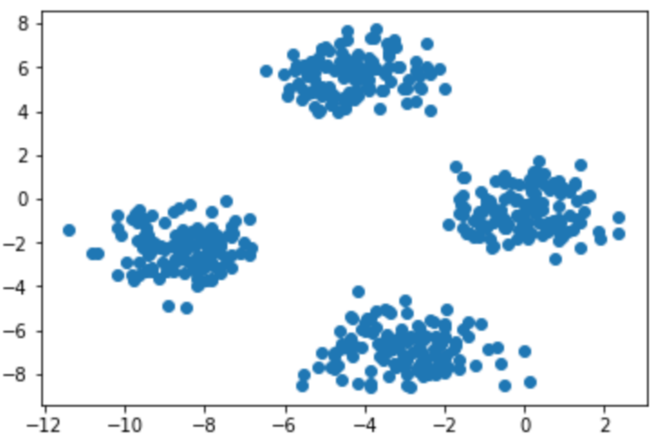


Figure 3.14: Plot of generated data

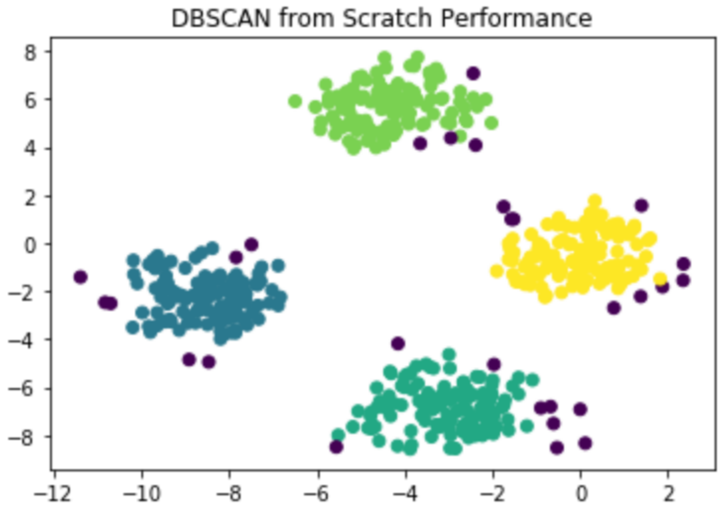


Figure 3.15: Plot of DBSCAN implementation

	OD_read	Proline
0	3.92	1065.0
1	3.40	1050.0
2	3.17	1185.0
3	3.45	1480.0
4	2.93	735.0

Figure 3.16: First five rows of wine dataset

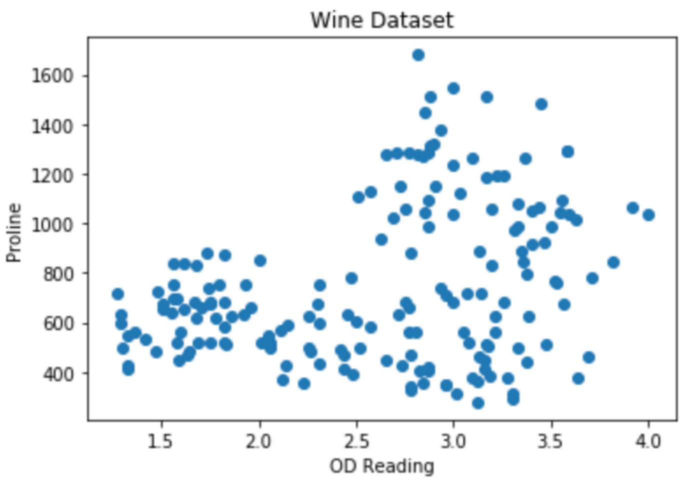


Figure 3.17: Plot of the data

```
Eps:  20 Min Samples:  5
DBSCAN Clustering:  0.3997987919957757
Eps:  25 Min Samples:  5
DBSCAN Clustering:  0.35258611037074095
Eps:  30 Min Samples:  5
DBSCAN Clustering:  0.43763797761597306
Eps:  25 Min Samples:  7
DBSCAN Clustering:  0.2711660466706248
Eps:  35 Min Samples:  7
DBSCAN Clustering:  0.4600630149335495
Eps:  35 Min Samples:  3
DBSCAN Clustering:  0.5368842164535846
```

Figure 3.18: Printing the silhouette score for clusters

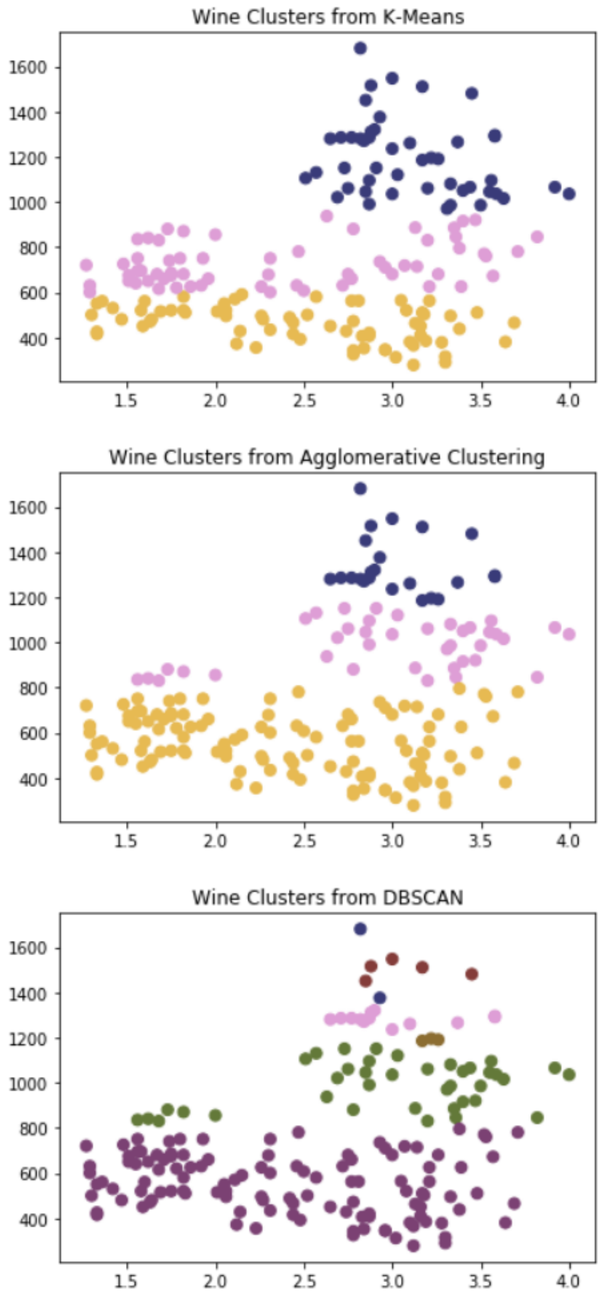


Figure 3.19: Plot of clusters using different algorithms

```
Silhouette Scores for Wine Dataset:

K-Means Clustering:  0.5809421087616886
Agg Clustering:  0.5988495817462
DBSCAN Clustering:  0.5368842164535846
```

Figure 3.20: Silhouette score

	Sepal Length	Sepal Width
0	5.1	3.5
1	4.9	3.0
2	4.7	3.2
3	4.6	3.1
4	5.0	3.6

Figure 4.43: The first five rows of the data

```
array([[ 0.68569351, -0.03926846],
       [-0.03926846,  0.18800403]])
```

Figure 4.44: The covariance matrix for the data

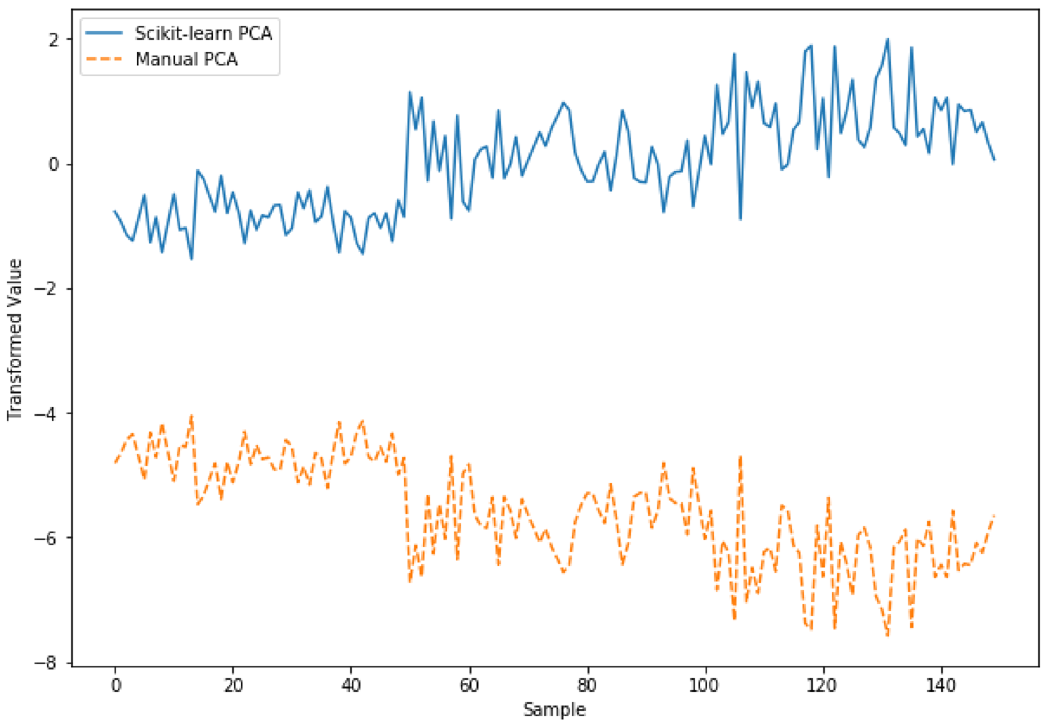


Figure 4.45: A plot of the data

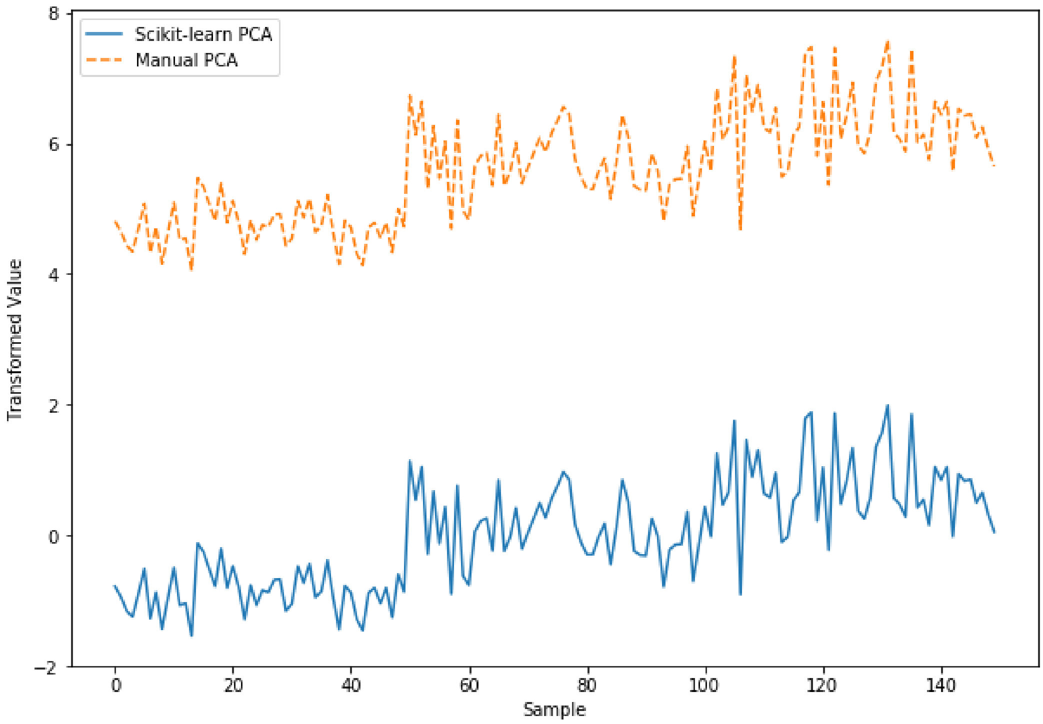


Figure 4.46: Re-plotted data

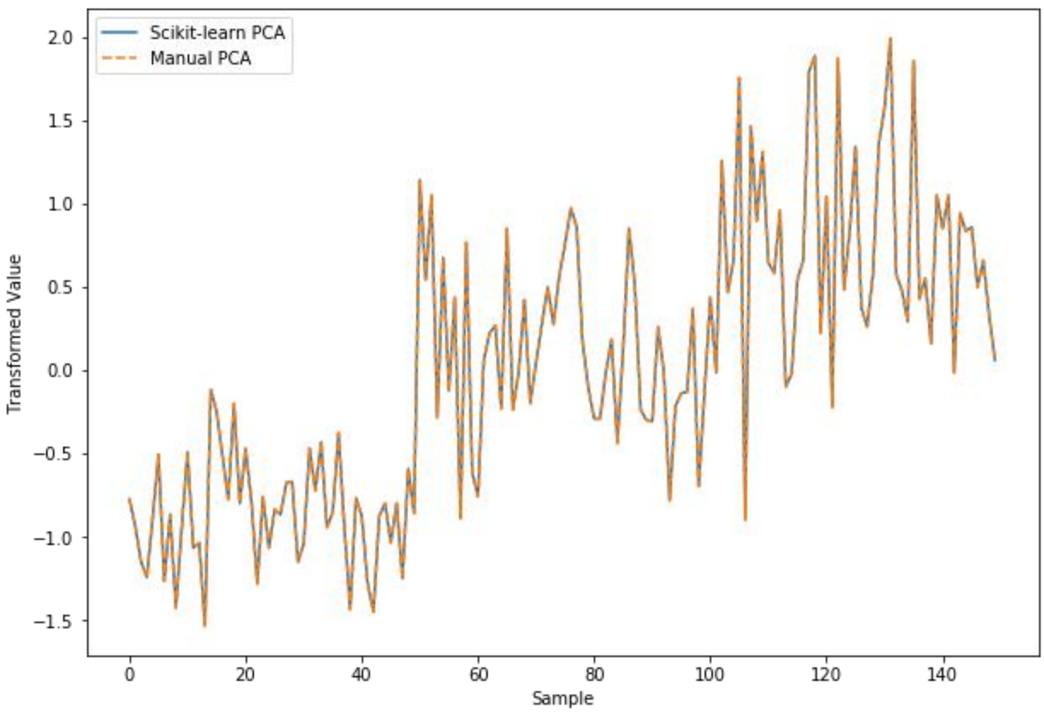


Figure 4.47: Re-plotting the data

	Sepal Length	Sepal Width	Petal Width
0	5.1	3.5	0.2
1	4.9	3.0	0.2
2	4.7	3.2	0.2
3	4.6	3.1	0.2
4	5.0	3.6	0.2

Figure 4.48: Sepal Length, Sepal Width, and Petal Width

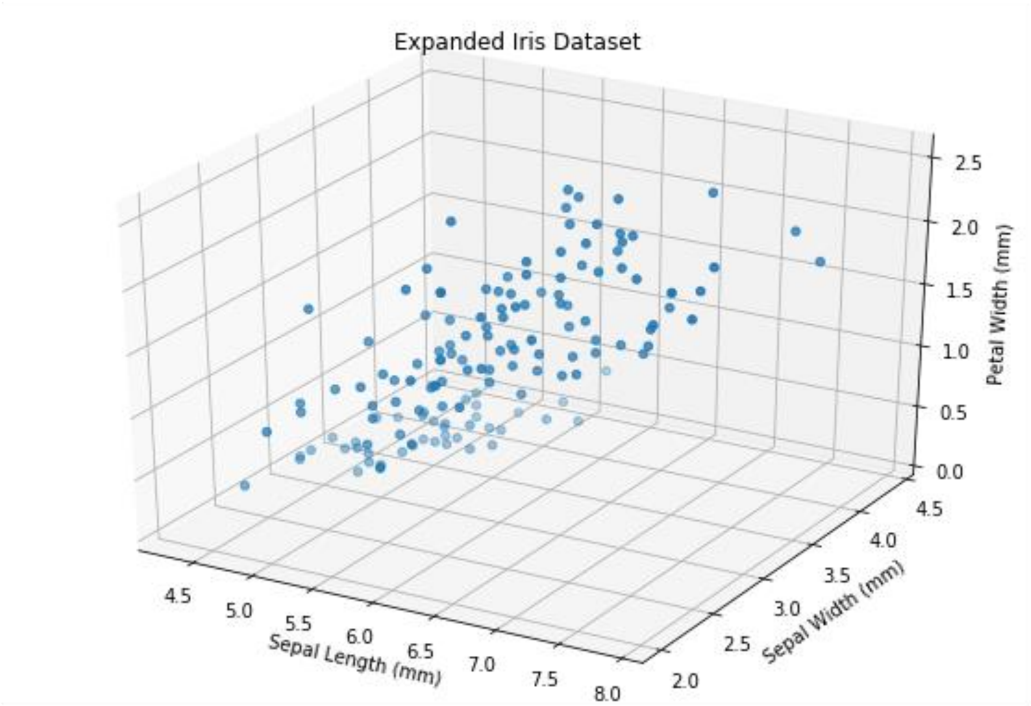


Figure 4.49: Expanded Iris dataset plot

```
PCA(copy=True, iterated_power='auto', n_components=None, random_state=None,
  svd_solver='auto', tol=0.0, whiten=False)
```

Figure 4.50: The model fitted to the dataset

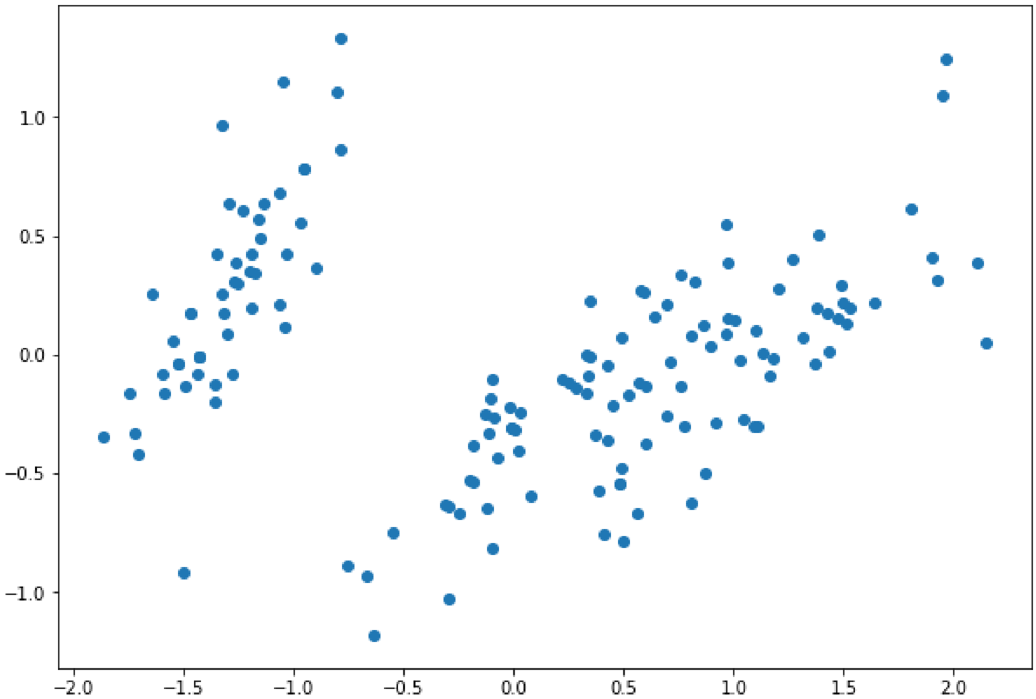


Figure 4.51: Plot of the transformed data

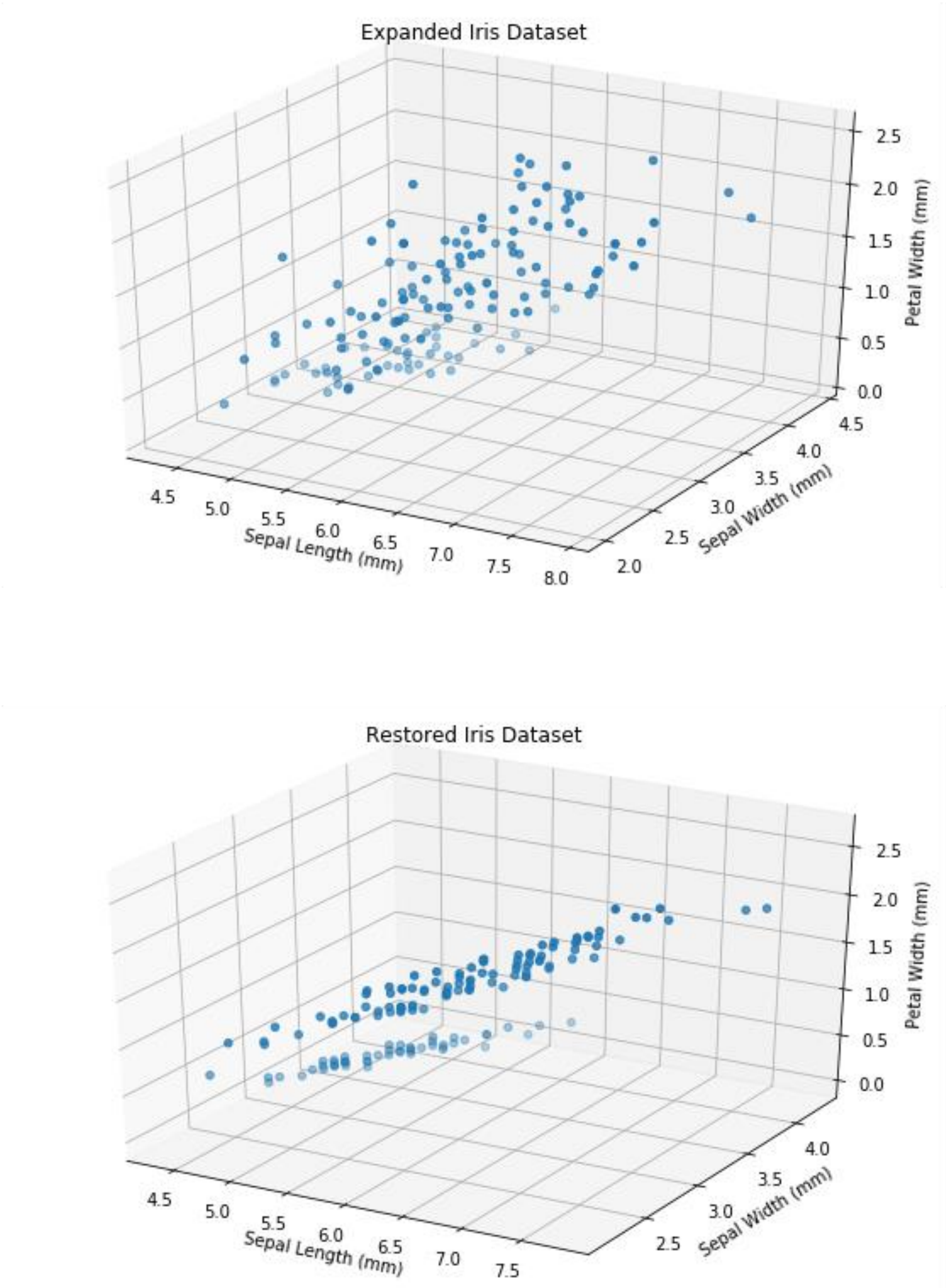


Figure 4.52: Plot of the expanded and the restored Iris datasets

```
array([-5.        , -4.8989899 , -4.7979798 , -4.6969697 , -4.5959596 ,
       -4.49494949, -4.39393939, -4.29292929, -4.19191919, -4.09090909,
       -3.98989899, -3.88888889, -3.78787879, -3.68686869, -3.58585859,
       -3.48484848, -3.38383838, -3.28282828, -3.18181818, -3.08080808,
       -2.97979798, -2.87878788, -2.77777778, -2.67676768, -2.57575758,
       -2.47474747, -2.37373737, -2.27272727, -2.17171717, -2.07070707,
       -1.96969697, -1.86868687, -1.76767677, -1.66666667, -1.56565657,
```

Figure 5.35: Printing the inputs

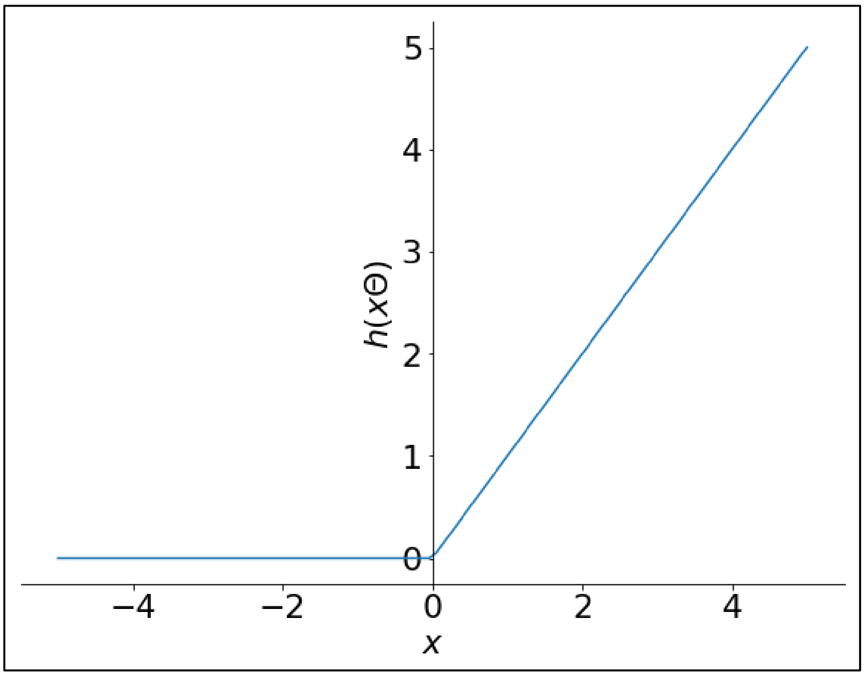


Figure 5.36: Plot of the neuron versus input

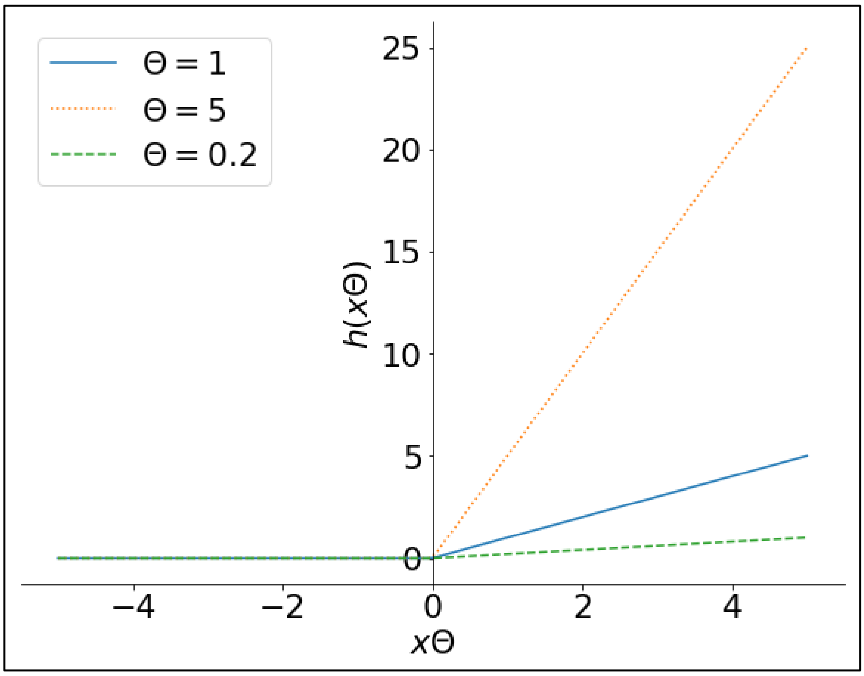


Figure 5.37: Three output curves of the neuron

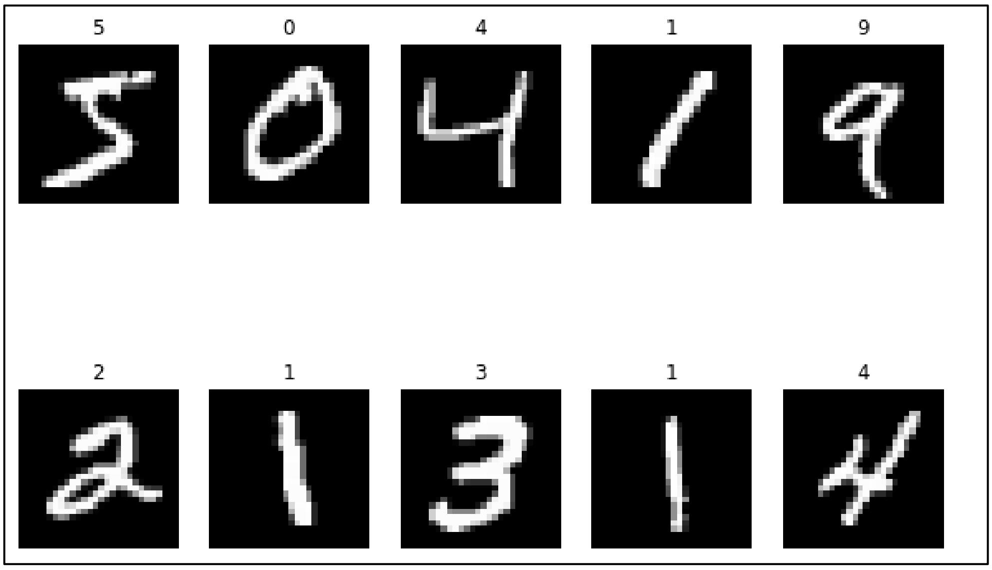


Figure 5.38: First 10 samples

```
array([[0., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 0.],
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 1., 0., 0.]])
```

Figure 5.39: Result of one hot encoding

```
10000/10000 [==============================] - 2s 152us/step - loss: 0.1963 - acc: 0.9471
Epoch 13/20
10000/10000 [==============================] - 2s 157us/step - loss: 0.1921 - acc: 0.9479
Epoch 14/20
10000/10000 [==============================] - 2s 173us/step - loss: 0.1877 - acc: 0.9487
Epoch 15/20
10000/10000 [==============================] - 2s 157us/step - loss: 0.1836 - acc: 0.9507
Epoch 16/20
10000/10000 [==============================] - 2s 156us/step - loss: 0.1791 - acc: 0.9522
Epoch 17/20
10000/10000 [==============================] - 2s 157us/step - loss: 0.1754 - acc: 0.9532
Epoch 18/20
10000/10000 [==============================] - 2s 158us/step - loss: 0.1714 - acc: 0.9538
Epoch 19/20
10000/10000 [==============================] - 2s 156us/step - loss: 0.1681 - acc: 0.9544
Epoch 20/20
10000/10000 [==============================] - 2s 160us/step - loss: 0.1638 - acc: 0.9559

<keras.callbacks.History at 0x7f60f7011f60>
```

Figure 5.40: Training the model

```
Epoch 96/100
10000/10000 [==============================] - 1s 130us/step - loss: 0.0755
Epoch 97/100
10000/10000 [==============================] - 1s 127us/step - loss: 0.0754
Epoch 98/100
10000/10000 [==============================] - 1s 126us/step - loss: 0.0754
Epoch 99/100
10000/10000 [==============================] - 1s 125us/step - loss: 0.0753
Epoch 100/100
10000/10000 [==============================] - 1s 128us/step - loss: 0.0752

<keras.callbacks.History at 0x7f5e9d2f0860>
```

Figure 5.41: Training the model

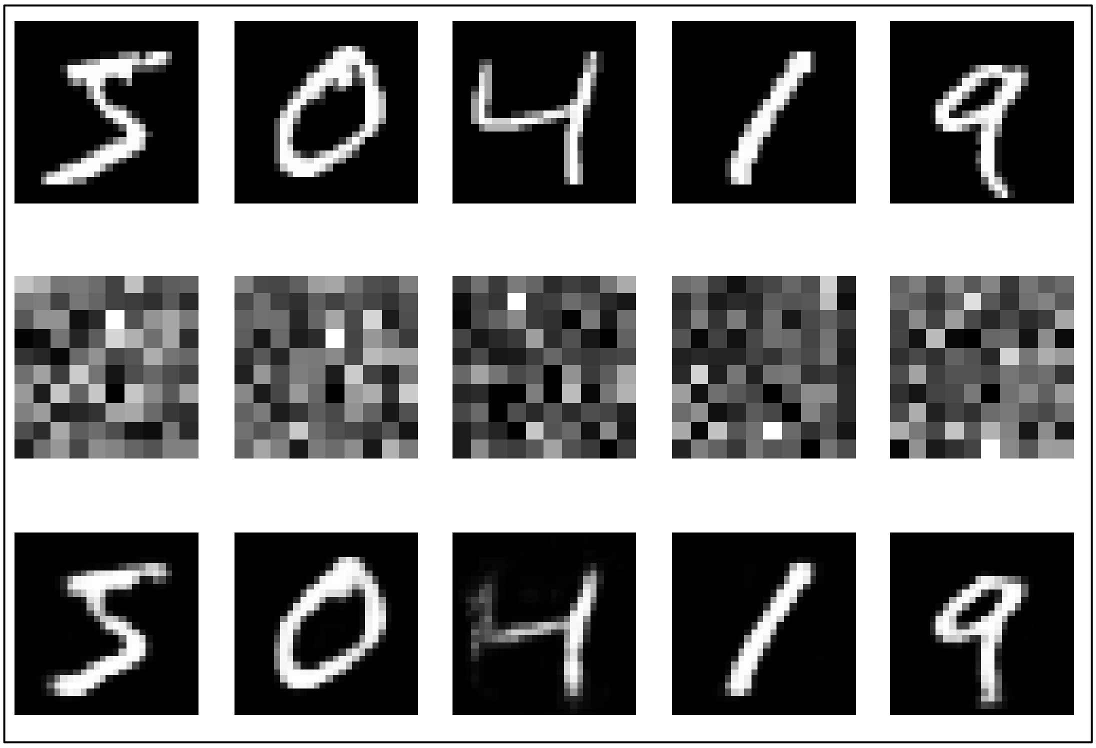


Figure 5.42: The original image, the encoder output, and the decoder

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 28, 28, 1)         0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 28, 28, 16)        160
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 14, 14, 16)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 14, 14, 16)        2320
_________________________________________________________________
up_sampling2d_1 (UpSampling2 (None, 28, 28, 16)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 28, 28, 1)         145
=================================================================
Total params: 2,625
Trainable params: 2,625
Non-trainable params: 0
```

Figure 5.43: Structure of model

```
Epoch 15/20
10000/10000 [==============================] - 9s 894us/step - loss: 0.0641
Epoch 16/20
10000/10000 [==============================] - 9s 931us/step - loss: 0.0640
Epoch 17/20
10000/10000 [==============================] - 9s 890us/step - loss: 0.0639
Epoch 18/20
10000/10000 [==============================] - 9s 943us/step - loss: 0.0638
Epoch 19/20
10000/10000 [==============================] - 9s 914us/step - loss: 0.0636
Epoch 20/20
10000/10000 [==============================] - 9s 931us/step - loss: 0.0635
```

Figure 5.44: Training the model

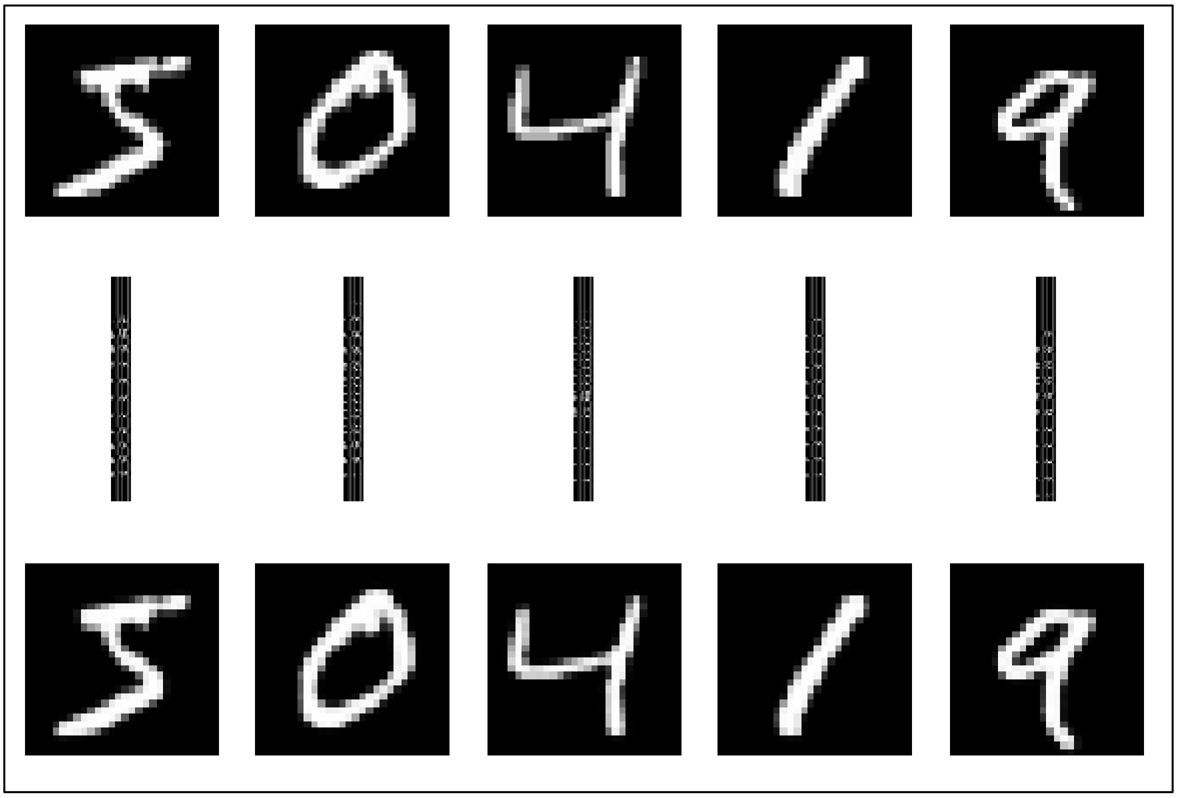


Figure 5.45: The original image, the encoder output, and the decoder

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

Figure 6.24: The first five rows of the wine dataset.

```
TSNE(angle=0.5, early_exaggeration=12.0, init='random', learning_rate=200.0,
    method='barnes_hut', metric='euclidean', min_grad_norm=1e-07,
    n_components=2, n_iter=1000, n_iter_without_progress=300,
    perplexity=30.0, random_state=0, verbose=1)
```

Figure 6.25: Creating t-SNE model.

```
[t-SNE] Computing 91 nearest neighbors...
[t-SNE] Indexed 178 samples in 0.000s...
[t-SNE] Computed neighbors for 178 samples in 0.003s...
[t-SNE] Computed conditional probabilities for sample 178 / 178
[t-SNE] Mean sigma: 9.207049
[t-SNE] KL divergence after 250 iterations with early exaggeration: 51.930435
[t-SNE] KL divergence after 900 iterations: 0.135609
```

Figure 6.26: Fitting PCA data t-SNE model

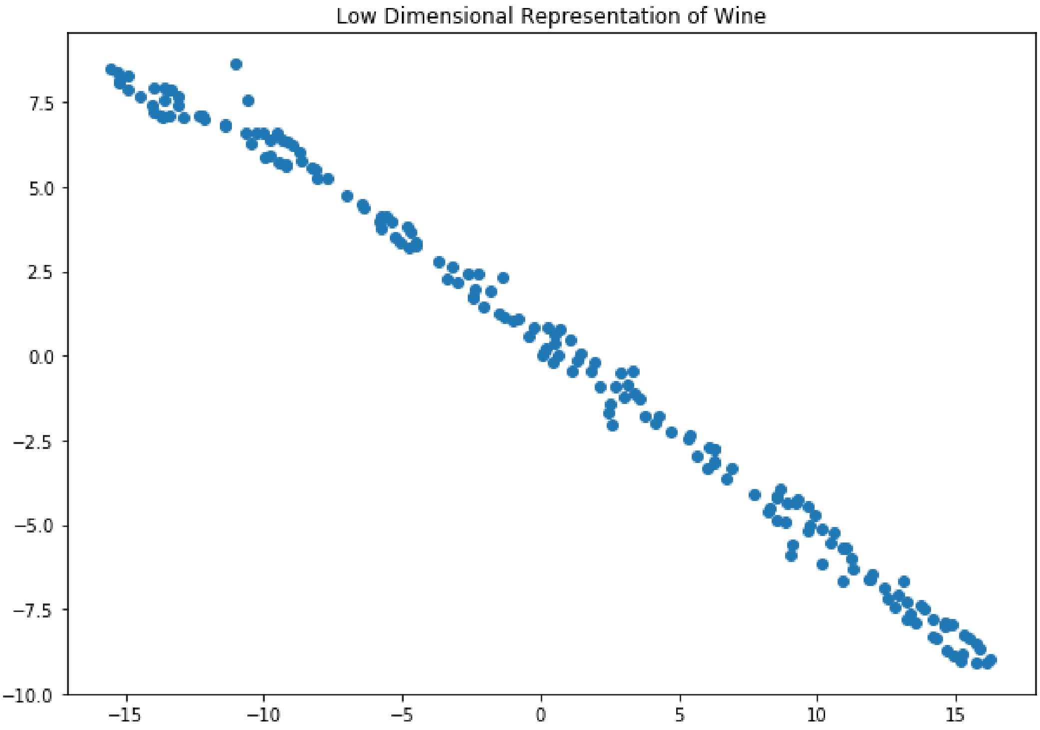


Figure 6.27: Scatterplot of two-dimensional data

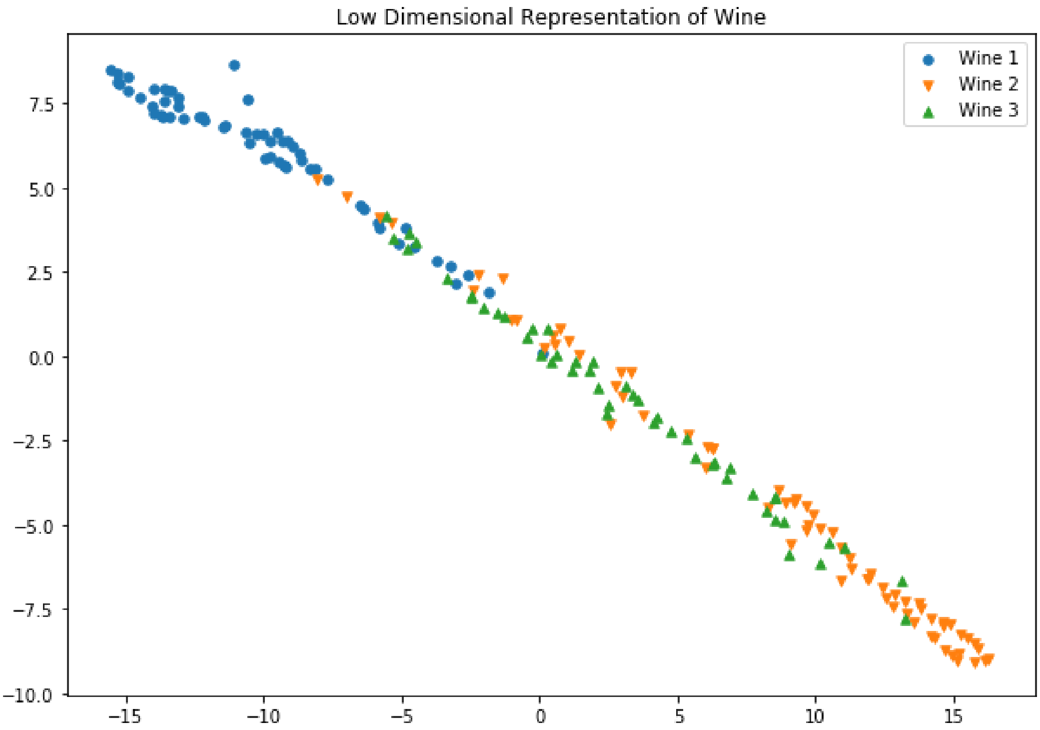


Figure 6.28: Secondary plot of two-dimensional data

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

Figure 6.29: The first five rows of wine data.

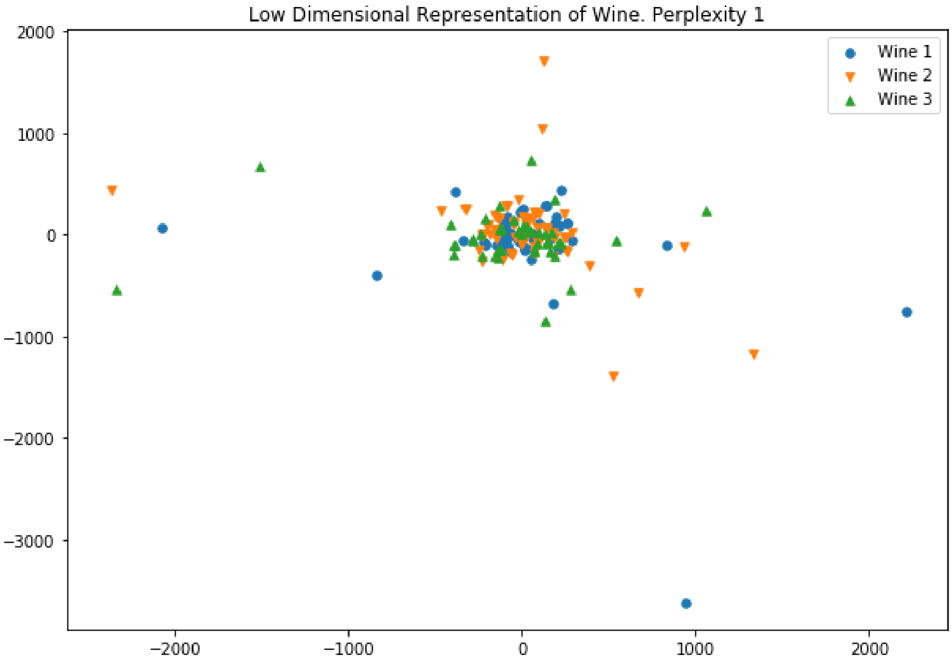


Figure 6.30: Plot for perplexity value 1

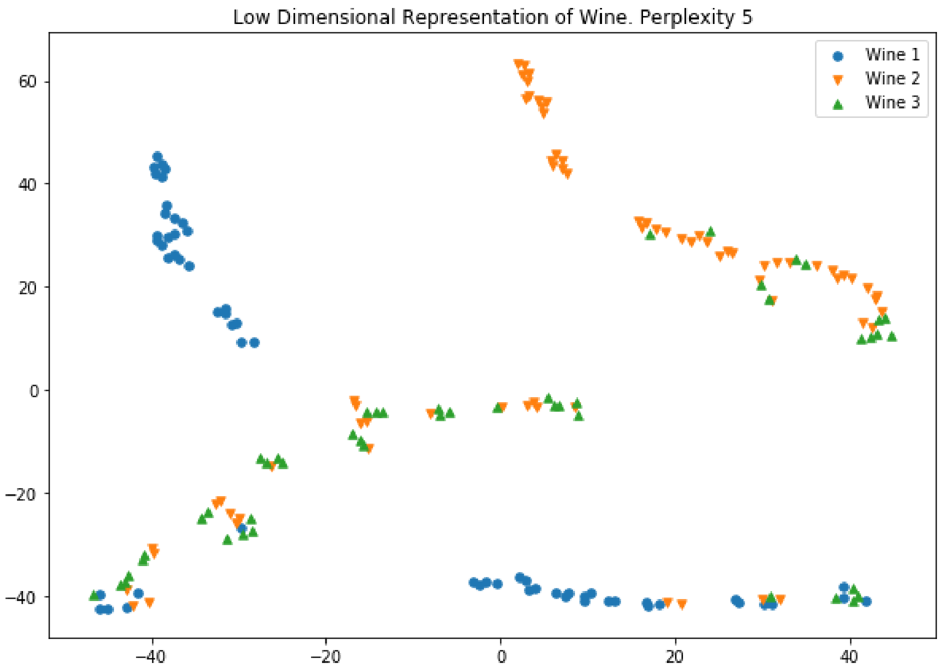


Figure 6.31: Plot for perplexity of 5

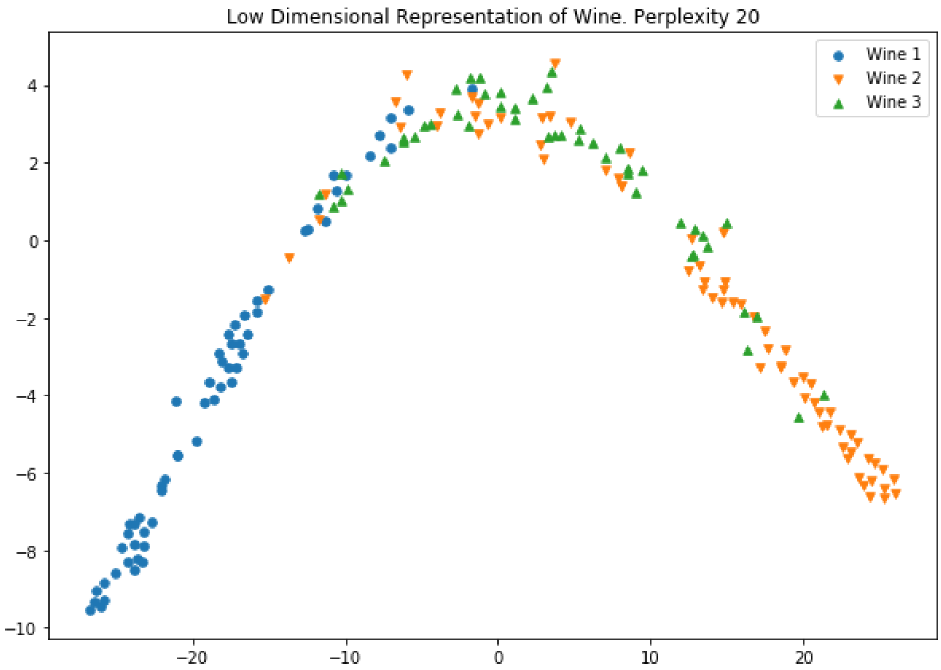


Figure 6.32: Plot for perplexity of 20

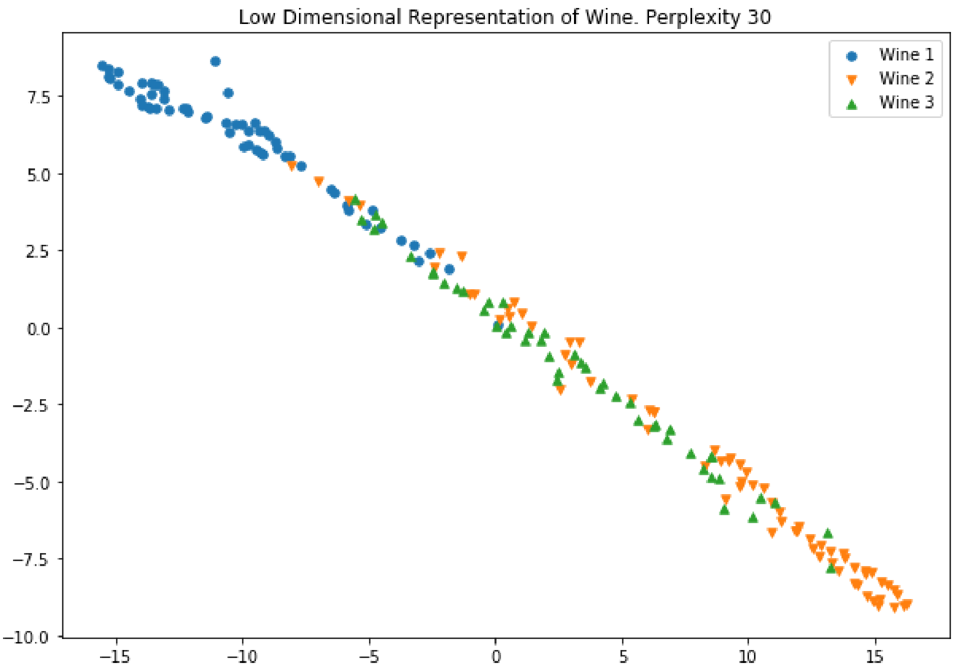


Figure 6.33: Plot for perplexity of 30

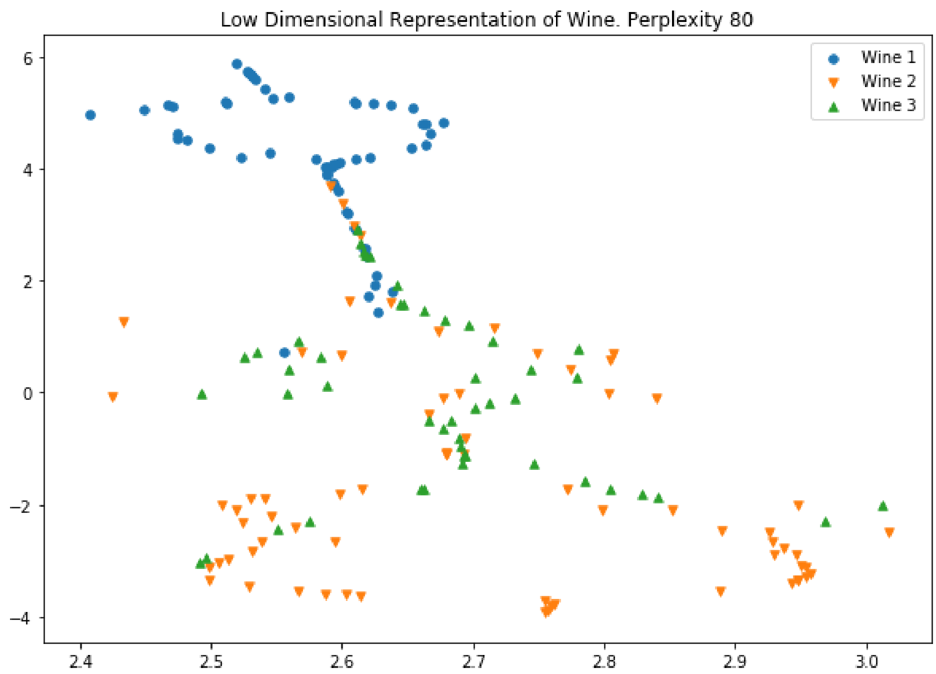


Figure 6.34: Plot for perplexity of 80

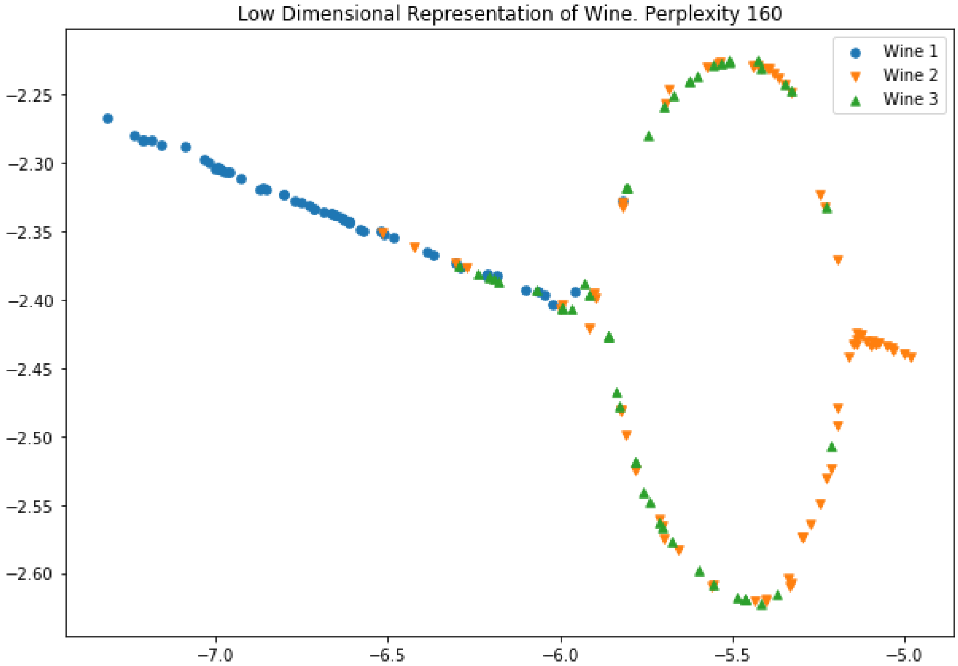


Figure 6.35: Plot for perplexity of 160

	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	1	14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64	1.04	3.92	1065
1	1	13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38	1.05	3.40	1050
2	1	13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68	1.03	3.17	1185
3	1	14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80	0.86	3.45	1480
4	1	13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32	1.04	2.93	735

Figure 6.36: The first five rows of wine dataset

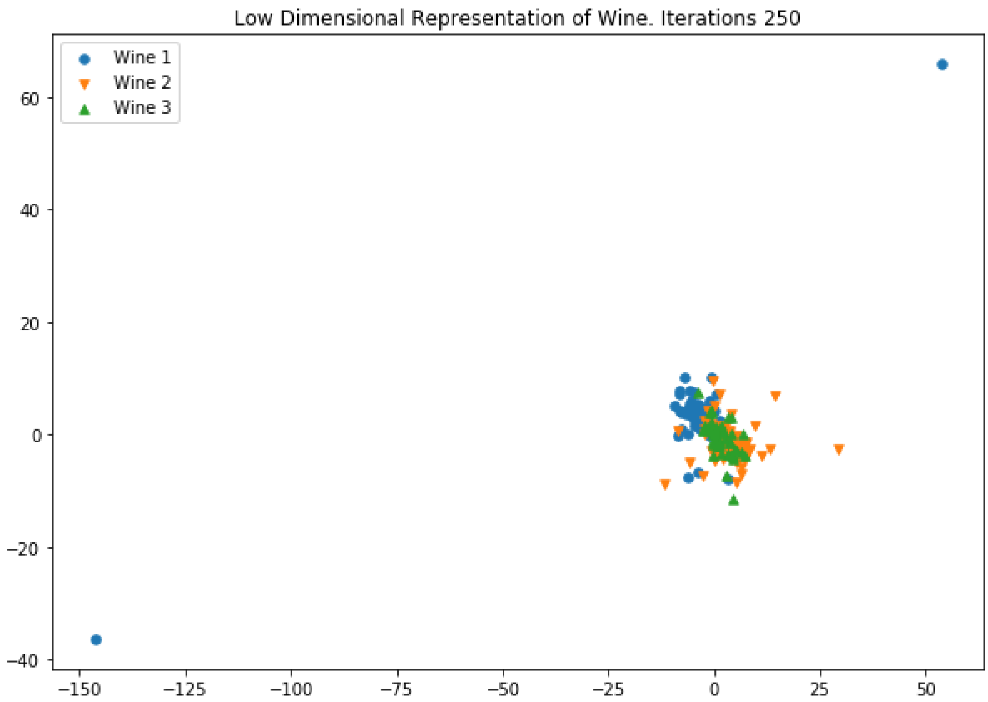


Figure 6.37: Scatterplot of wine classes with 250 iterations

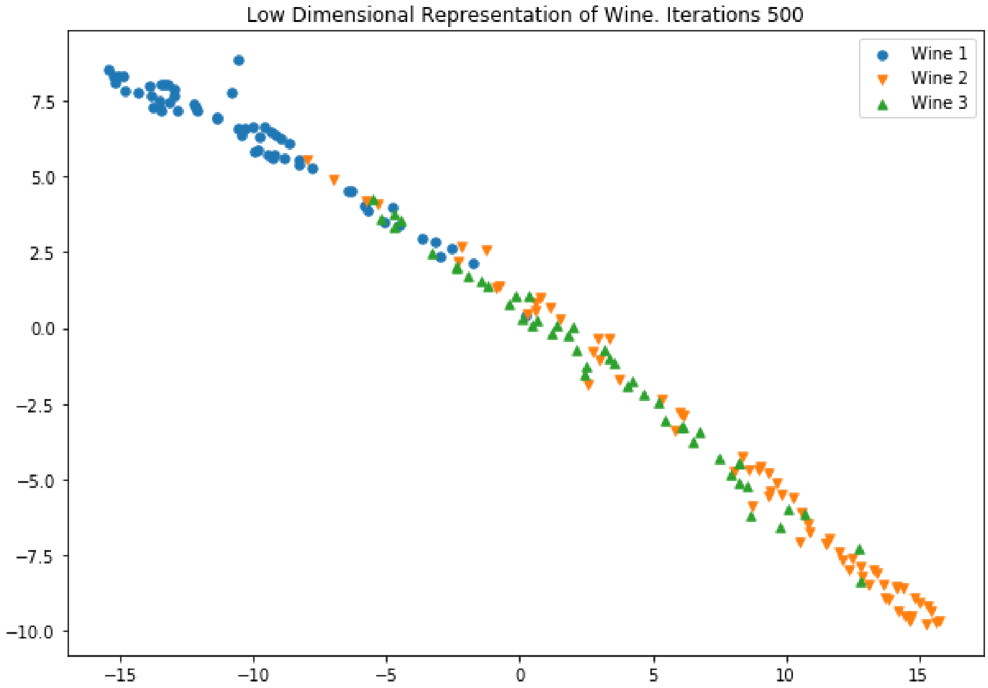


Figure 6.38: Scatterplot of wine classes with 500 iterations

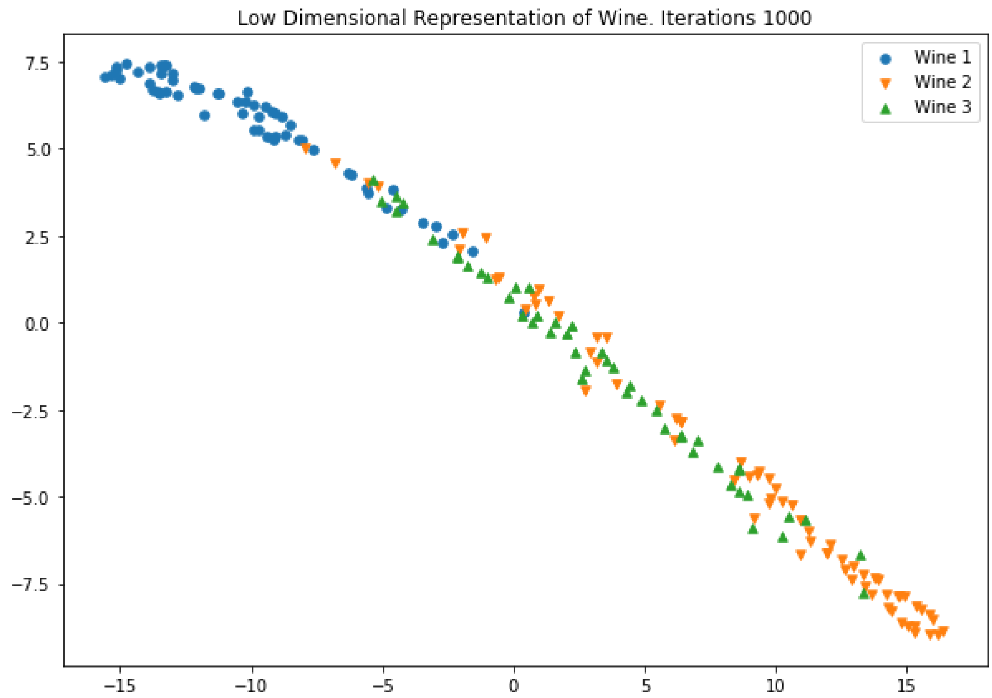


Figure 6.39: Scatterplot of wine classes with 1,000 iterations

```
SHAPE:
(4171, 3)

COLUMN NAMES:
Index(['id', 'datetime', 'tweettext'], dtype='object')

HEAD:
                   id                         datetime  \
0  576760256031682561  Sat Mar 14 15:02:15 +0000 2015
1  576715414811471872  Sat Mar 14 12:04:04 +0000 2015

                                    tweettext
0  Five new running shoes that aim to go the extr...
1  Gym Rat: Disq class at Crunch is intense worko...
```

Figure 7.54: Shape, column names, and head of data

```
HEADLINES:
['Five new running shoes that aim to go the extra mile http://lat.ms/1ELp3wU', 'Gym Rat: Disq class at Crunch is intense workou
t on pulley system http://lat.ms/1EKOFdr', 'Noshing through thousands of ideas at Natural Products Expo West http://lat.ms/1EHq
ywg', 'Natural Products Expo also explores beauty, supplements and more http://lat.ms/1EHqyfE', 'Free Fitness Weekends in South
Bay beach cities aim to spark activity http://lat.ms/1EH3SMC']

LENGTH:
4171
```

Figure 7.55: Headlines and their length

```
HEADLINES:
[['running', 'shoes', 'extra'], ['class', 'crunch', 'intense', 'workout', 'pulley', 'system'], ['thousand', 'natural', 'produc
t'], ['natural', 'product', 'explore', 'beauty', 'supplement'], ['fitness', 'weekend', 'south', 'beach', 'spark', 'activity']]

LENGTH:
4093
```

Figure 7.56: Headline and length after removing None

```
['running shoes extra', 'class crunch intense workout pulley system', 'thousand natural product', 'natural product explore beau
ty supplement', 'fitness weekend south beach spark activity', 'kayla harrison sacrifice', 'sonic treatment alzheimers disease',
'ultrasound brain restore memory alzheimers needle onlyso farin mouse', 'apple researchkit really medical research', 'warning c
hantix drink taking might remember']
```

Figure 7.57: Tweets cleaned for modeling

	Number Of Topics	Perplexity Score
0	2	349.004885
1	4	404.137619
2	6	440.677441
3	8	464.222793
4	10	478.094739
5	12	493.116250
6	14	506.144776
7	16	524.674504
8	18	530.975575
9	20	535.461393

Figure 7.58: Number of topics versus perplexity score data frame

```
LatentDirichletAllocation(batch_size=128, doc_topic_prior=None,
             evaluate_every=-1, learning_decay=0.7,
             learning_method='online', learning_offset=10.0,
             max_doc_update_iter=100, max_iter=10, mean_change_tol=0.001,
             n_components=2, n_jobs=None, n_topics=None, perp_tol=0.1,
             random_state=0, topic_word_prior=None,
             total_samples=1000000.0, verbose=0)
```

Figure 7.59: LDA model

	Topic0	Topic1
Word0	(0.0417, latfit)	(0.0817, study)
Word1	(0.0336, health)	(0.0306, cancer)
Word2	(0.0242, people)	(0.0212, patient)
Word3	(0.0203, could)	(0.0172, death)
Word4	(0.0192, brain)	(0.017, obesity)
Word5	(0.018, researcher)	(0.0168, doctor)
Word6	(0.0176, woman)	(0.0166, heart)
Word7	(0.016, report)	(0.0148, disease)
Word8	(0.0143, california)	(0.0144, weight)
Word9	(0.0125, scientist)	(0.0115, research)

Figure 7.60: Word-topic table for the health tweet data

```
                                                  Topic0  \
Doc0  (0.9443, Want your legs to look good in those ...
Doc1  (0.9442, 11% of hospital patients got care the...
Doc2  (0.9373, Spend time with dad this Father’s Day...
Doc3  (0.9373, Hve fun! That's an order. It's import...
Doc4  (0.9372, Need a new challenge for your ab work...
Doc5  (0.9368, ZMapp goes 18-for-18 in treating monk...
Doc6  (0.9367, Anti-vaccination activists target hig...
Doc7  (0.9337, RT @latimesscience: @xprize pulled th...
Doc8  (0.9285, About 75% of homeless people smoke, a...
Doc9  (0.9284, Yogi crunches can give you flat abs a...

                                                  Topic1
Doc0  (0.9498, Computer problems are delaying nursin...
Doc1  (0.9457, Trans fats? DONE. Will the @US_FDA go...
Doc2  (0.9414, Supplements to boost "low T" increase...
Doc3  (0.9372, Study: The 2009 H1N1 "swine flu" pand...
Doc4  (0.9363, Doctors often delay vaccines for youn...
Doc5  (0.9357, Humans eat more calories, protein and...
Doc6  (0.9356, Las Vegas: Finding the latest in bike...
Doc7  (0.9354, Soccer players' ACL injury risk may d...
Doc8  (0.9284, Men walk more slowly with a woman IF ...
Doc9  (0.9284, Do blood transfusions from Ebola surv...
```

Figure 7.61: Document topic table

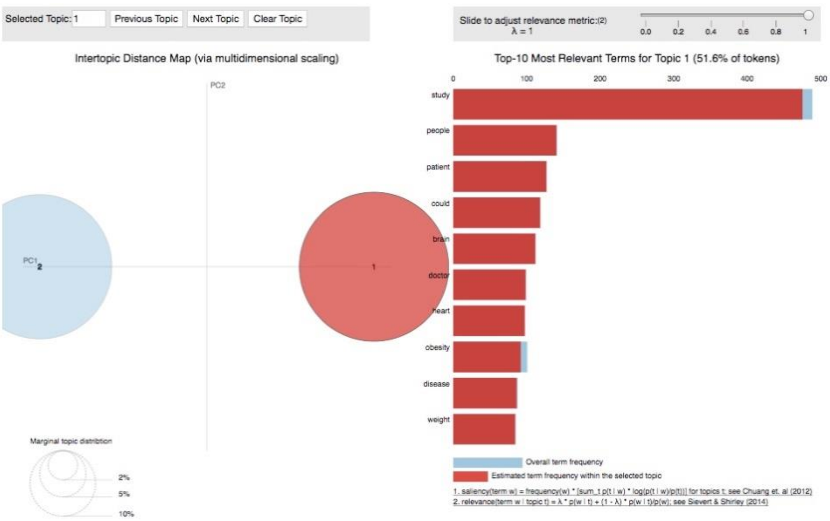


Figure 7.62: A histogram and biplot for the LDA model trained on health tweets

```
NMF(alpha=0.1, beta_loss='frobenius', init='nndsvda', l1_ratio=0.5,
  max_iter=200, n_components=2, random_state=0, shuffle=False, solver='mu',
  tol=0.0001, verbose=0)
```

Figure 7.63: Defining the NMF model

	Topic0	Topic1
Word0	(0.3794, study)	(0.5955, latfit)
Word1	(0.0256, cancer)	(0.0487, steps)
Word2	(0.0207, people)	(0.0446, today)
Word3	(0.0183, obesity)	(0.0402, exercise)
Word4	(0.0183, brain)	(0.0273, healthtips)
Word5	(0.0182, health)	(0.0258, workout)
Word6	(0.0175, suggest)	(0.0203, getting)
Word7	(0.0167, weight)	(0.0192, fitness)
Word8	(0.0152, woman)	(0.0143, great)
Word9	(0.013, death)	(0.0131, morning)

Figure 7.64: The word-topic table with probabilities

	6 CHOCOLATE LOVE HEART T-LIGHTS	6 EGG HOUSE PAINTED WOOD	6 GIFT TAGS 50'S CHRISTMAS	6 GIFT TAGS VINTAGE CHRISTMAS	6 RIBBONS ELEGANT CHRISTMAS	6 RIBBONS EMPIRE	6 RIBBONS RUSTIC CHARM	6 RIBBONS SHIMMERING PINKS	6 ROCKET BALLOONS	60 CAKE CASES DOLLY GIRL DESIGN
20125	False	False	False	False	False	False	False	False	False	False
20126	False	False	False	False	False	False	False	False	False	False
20127	False	False	False	False	False	False	False	False	False	False
20128	False	False	False	False	False	False	False	False	False	False
20129	False	False	False	False	False	False	False	False	False	False
20130	False	False	False	False	False	False	False	False	False	False
20131	False	False	False	False	False	False	False	False	False	False
20132	False	False	False	False	False	False	False	False	False	False
20133	False	False	False	False	False	False	False	False	False	False
20134	False	False	False	False	False	False	False	False	False	False
20135	False	False	False	False	False	False	False	False	False	False

Figure 8.35: A subset of the cleaned, encoded, and recast DataFrame built from the complete online retail dataset

	support	itemsets
0	0.013359	(SET 2 TEA TOWELS I LOVE LONDON)
1	0.015793	(10 COLOUR SPACEBOY PEN)
2	0.012465	(12 MESSAGE CARDS WITH ENVELOPES)
3	0.017630	(12 PENCIL SMALL TUBE WOODLAND)
4	0.017978	(12 PENCILS SMALL TUBE RED RETROSPOT)
5	0.017630	(12 PENCILS SMALL TUBE SKULL)
6	0.013309	(12 PENCILS TALL TUBE RED RETROSPOT)

Figure 8.36: The Apriori algorithm results using the complete online retail dataset

	support	itemsets
1	0.015793	(10 COLOUR SPACEBOY PEN)

Figure 8.37: Result of item set containing 10 COLOUR SPACEBOY PEN

	support	itemsets	length
836	0.020759	(ALARM CLOCK BAKELIKE PINK, ALARM CLOCK BAKELI...	2
887	0.020362	(CHARLOTTE BAG SUKI DESIGN, CHARLOTTE BAG PINK...	2
923	0.020610	(CHARLOTTE BAG SUKI DESIGN, STRAWBERRY CHARLOT...	2
1105	0.020560	(JUMBO BAG PINK POLKADOT, JUMBO BAG BAROQUE B...	2
1114	0.020908	(JUMBO SHOPPER VINTAGE RED PAISLEY, JUMBO BAG...	2
1116	0.020957	(JUMBO STORAGE BAG SUKI, JUMBO BAG BAROQUE BL...	2
1129	0.020560	(JUMBO BAG RED RETROSPOT, JUMBO BAG ALPHABET)	2
1137	0.020163	(JUMBO BAG PEARS, JUMBO BAG APPLES)	2
1203	0.020709	(JUMBO SHOPPER VINTAGE RED PAISLEY, JUMBO BAG ...	2
1218	0.020560	(JUMBO STORAGE BAG SKULLS, JUMBO BAG RED RETRO...	2
1236	0.020610	(RECYCLING BAG RETROSPOT , JUMBO BAG RED RETRO...	2
1328	0.020610	(LUNCH BAG BLACK SKULL., LUNCH BAG APPLE DESIGN)	2
1390	0.020610	(LUNCH BAG SUKI DESIGN , LUNCH BAG PINK POLKADOT)	2
1458	0.020610	(WHITE HANGING HEART T-LIGHT HOLDER, NATURAL S...	2

Figure 8.38: The section of the results of filtering based on length and support

```
Text(0.5, 1.0, 'Support')
```

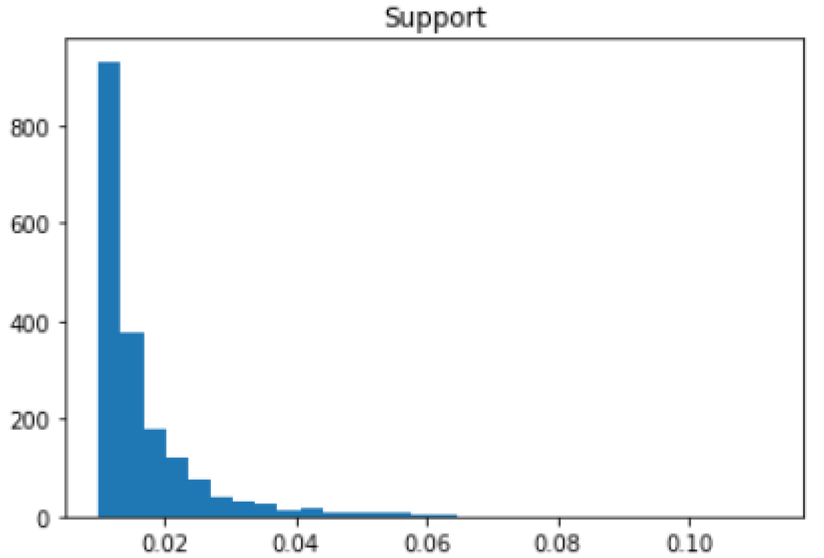


Figure 8.39: The distribution of support values

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(ALARM CLOCK BAKELIKE CHOCOLATE)	(ALARM CLOCK BAKELIKE GREEN)	0.021255	0.048669	0.013756	0.647196	13.297902	0.012722	2.696488
1	(ALARM CLOCK BAKELIKE CHOCOLATE)	(ALARM CLOCK BAKELIKE RED)	0.021255	0.052195	0.014501	0.682243	13.071023	0.013392	2.982798
2	(ALARM CLOCK BAKELIKE ORANGE)	(ALARM CLOCK BAKELIKE GREEN)	0.022100	0.048669	0.013558	0.613483	12.605201	0.012482	2.461292
3	(ALARM CLOCK BAKELIKE RED)	(ALARM CLOCK BAKELIKE GREEN)	0.052195	0.048669	0.031784	0.608944	12.511932	0.029244	2.432722
4	(ALARM CLOCK BAKELIKE GREEN)	(ALARM CLOCK BAKELIKE RED)	0.048669	0.052195	0.031784	0.653061	12.511932	0.029244	2.731908
5	(ALARM CLOCK BAKELIKE IVORY)	(ALARM CLOCK BAKELIKE RED)	0.028308	0.052195	0.018524	0.654386	12.537313	0.017047	2.742380
6	(ALARM CLOCK BAKELIKE ORANGE)	(ALARM CLOCK BAKELIKE RED)	0.022100	0.052195	0.014998	0.678652	13.002217	0.013845	2.949463

Figure 8.40: The association rules based on the complete online retail dataset

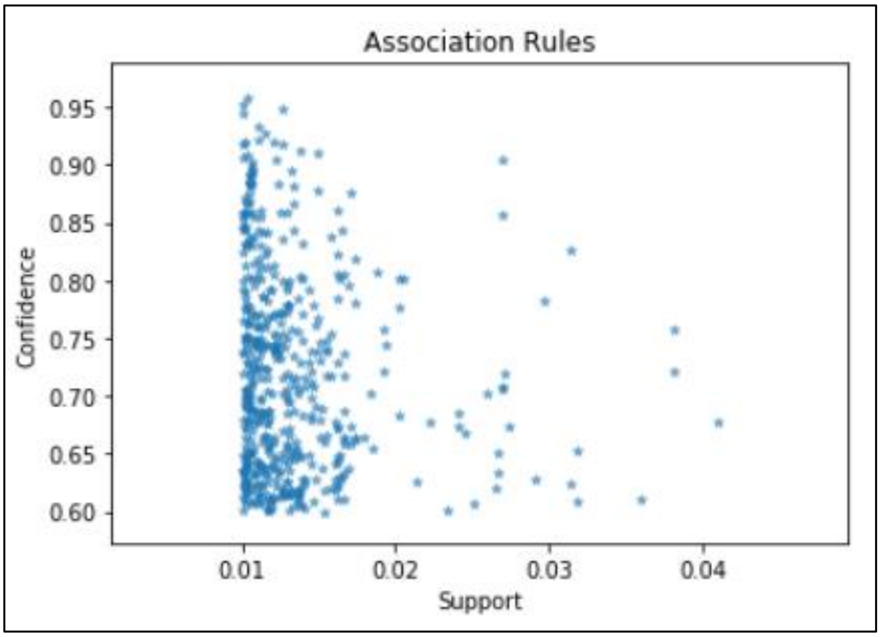


Figure 8.41: The plot of confidence against support

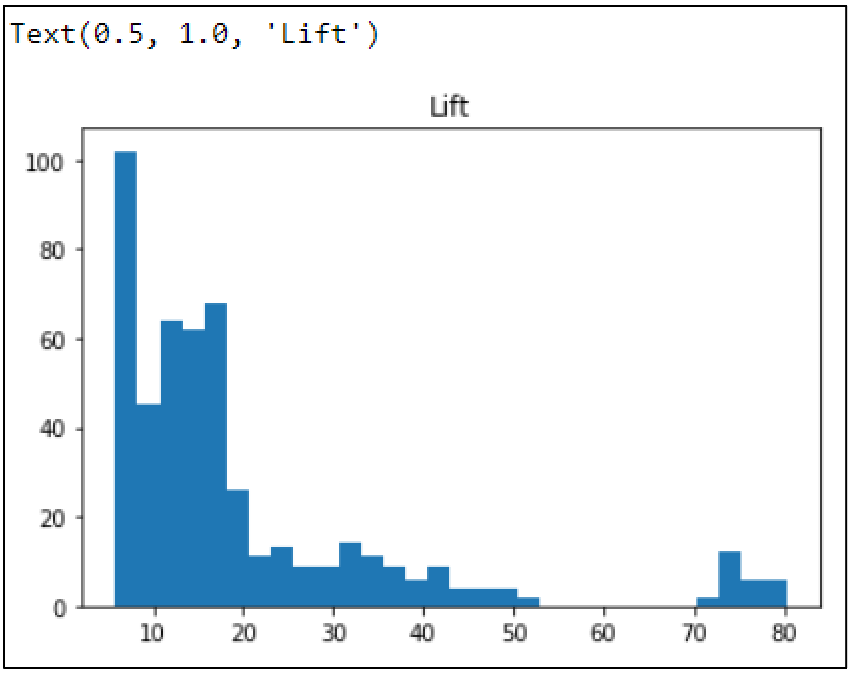


Figure 8.42: The distribution of lift values

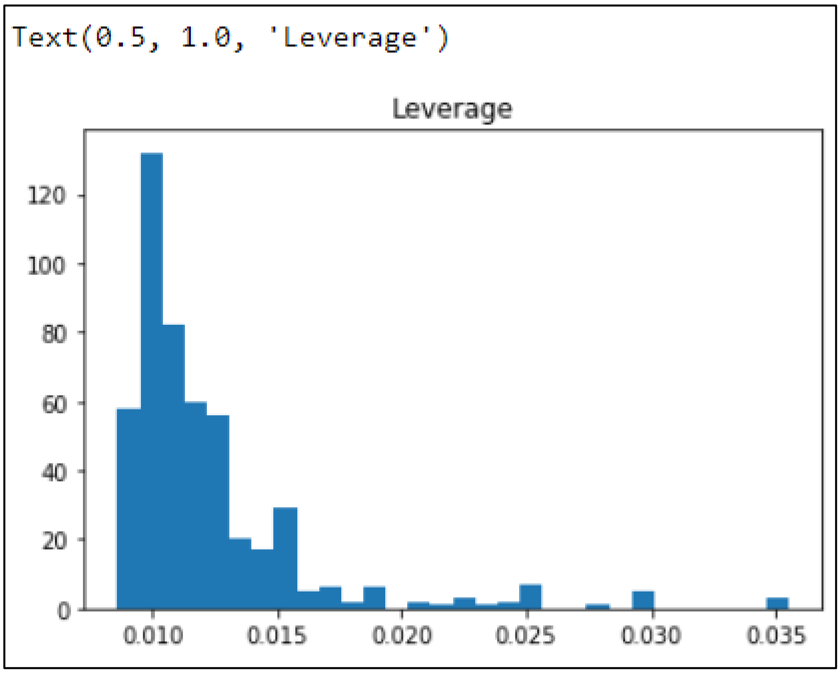


Figure 8.43: The distribution of leverage values

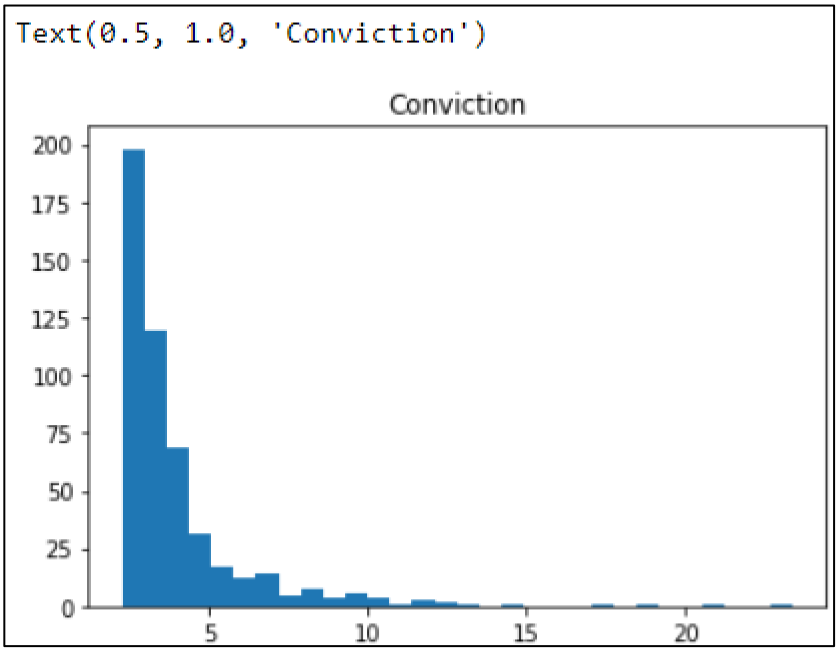


Figure 8.44: The distribution of conviction values

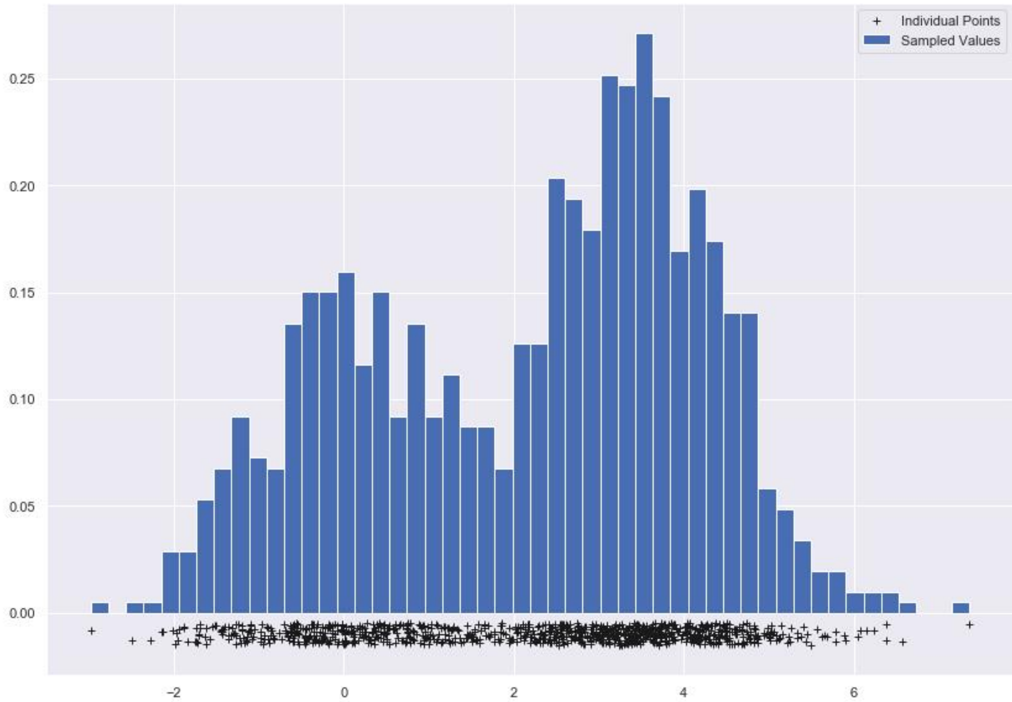


Figure 9.29: A histogram of the random sample with a scatterplot underneath

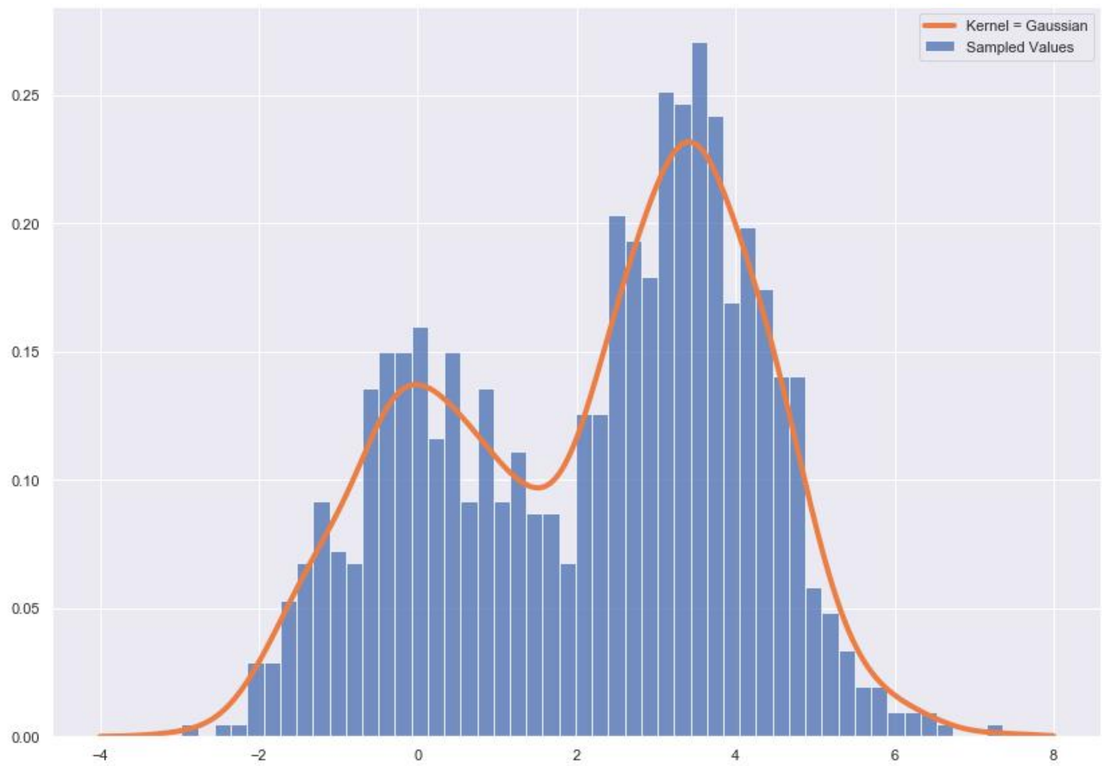


Figure 9.30: A histogram of the random sample with the optimal estimated density overlaid

```
Month: 2018-07
Dimensions: (95677, 12)
Head:
                                            Crime ID    Month  \
0  e9fe81ec7a6f5d2a80445f04be3d7e92831dbf3090744e...  2018-07
1  076b796ba1e1ba3f69c9144e2aa7a7bc85b61d51bf7a59...  2018-07

                   Reported by                 Falls within  Longitude  \
0  Metropolitan Police Service  Metropolitan Police Service   0.774271
1  Metropolitan Police Service  Metropolitan Police Service  -1.007293

    Latitude                     Location  LSOA code           LSOA name  \
0  51.148147  On or near Bethersden Road  E01024031         Ashford 012B
1  51.893136           On or near Prison  E01017674  Aylesbury Vale 010D

    Crime type      Last outcome category  Context
0  Other theft  Status update unavailable      NaN
1  Other crime     Awaiting court outcome      NaN
```

Figure 9.31: An example of one of the individual crime files

```
Dimensions - Full Data:
(546032, 12)

Unique Months - Full Data:
['2018-07' '2018-08' '2018-09' '2018-10' '2018-11' '2018-12']

Number of Unique Crime Types - Full Data:
14

Unique Crime Types - Full Data:
['Other theft' 'Other crime' 'Violence and sexual offences'
 'Anti-social behaviour' 'Criminal damage and arson' 'Drugs'
 'Possession of weapons' 'Theft from the person' 'Vehicle crime'
 'Burglary' 'Public order' 'Robbery' 'Shoplifting' 'Bicycle theft']

Count Occurrences Of Each Unique Crime Type - Full Type:
Violence and sexual offences    117499
Anti-social behaviour           115448
Other theft                      61833
Vehicle crime                    58857
Burglary                         41145
Criminal damage and arson        28436
Public order                     24655
Theft from the person            22670
Shoplifting                      21296
Drugs                            17292
Robbery                          17060
Bicycle theft                    11362
Other crime                       5223
Possession of weapons             3256
Name: Crime type, dtype: int64
```

Figure 9.32: Descriptors of the full crime dataset

	Month	Longitude	Latitude	Crime type
0	2018-07	0.774271	51.148147	Other theft
1	2018-07	-1.007293	51.893136	Other crime
2	2018-07	0.744706	52.038219	Violence and sexual offences
3	2018-07	0.148434	51.595164	Anti-social behaviour
4	2018-07	0.137065	51.583672	Anti-social behaviour

Figure 9.33: Crime data in DataFrame form subset down to the Longitude, Latitude, Month, and Crime type columns

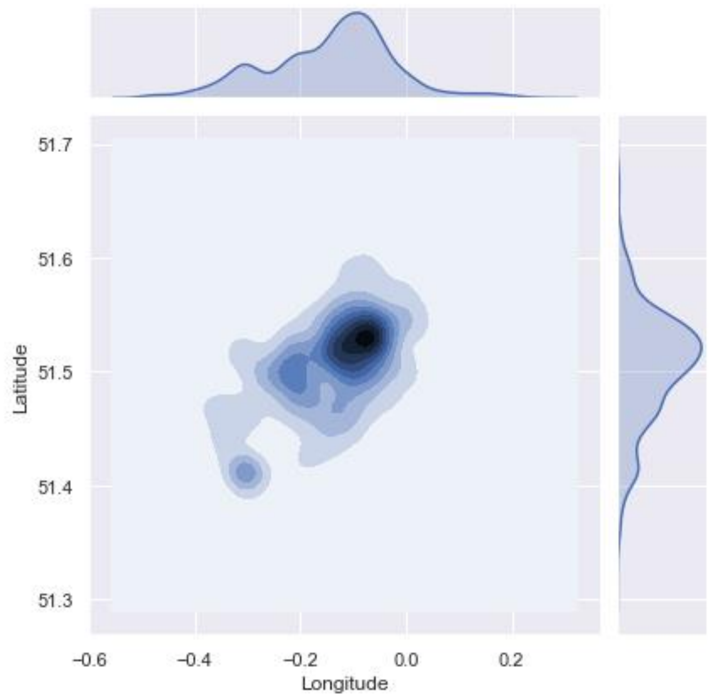


Figure 9.34: The estimated joint and marginal densities for bicycle thefts in July 2018

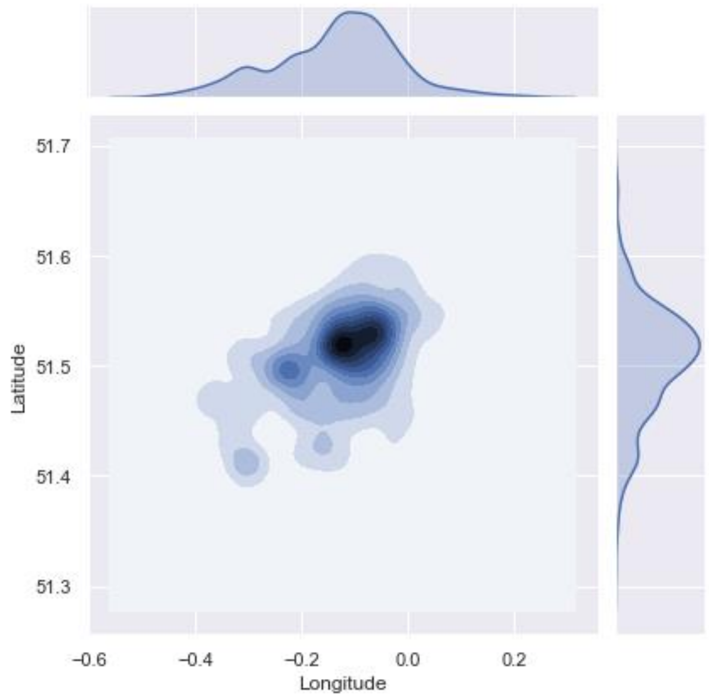


Figure 9.35: The estimated joint and marginal densities for bicycle thefts in September 2018

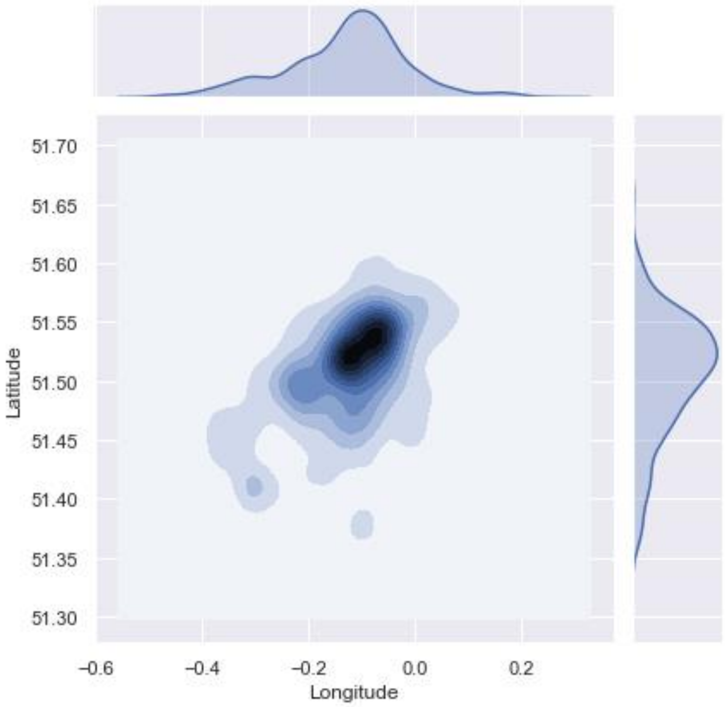


Figure 9.36: The estimated joint and marginal densities for bicycle thefts in December 2018

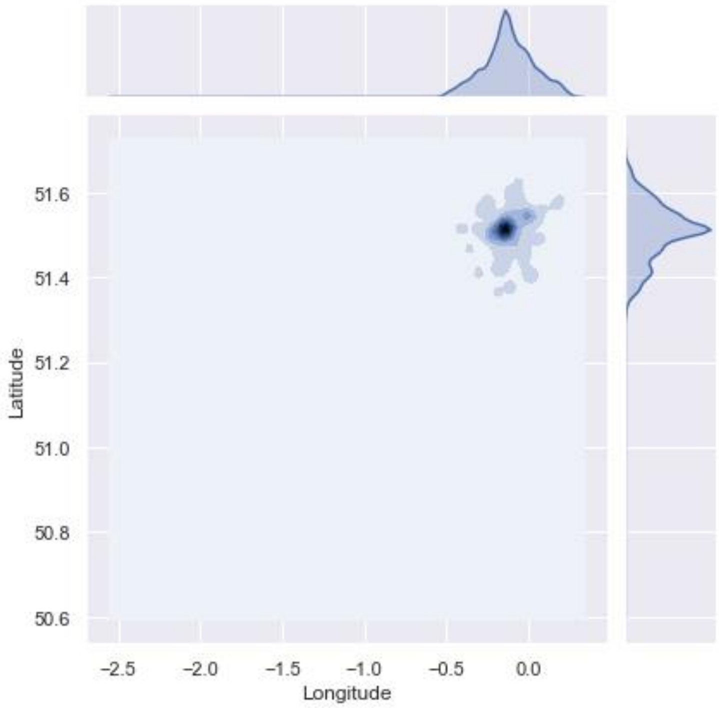


Figure 9.37: The estimated joint and marginal densities for shoplifting incidents in August 2018

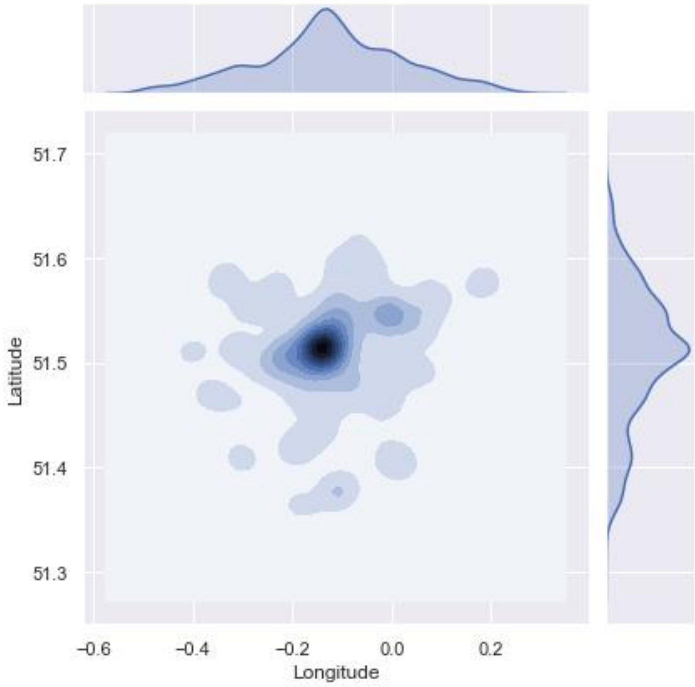


Figure 9.38: The estimated joint and marginal densities for shoplifting incidents in October 2018

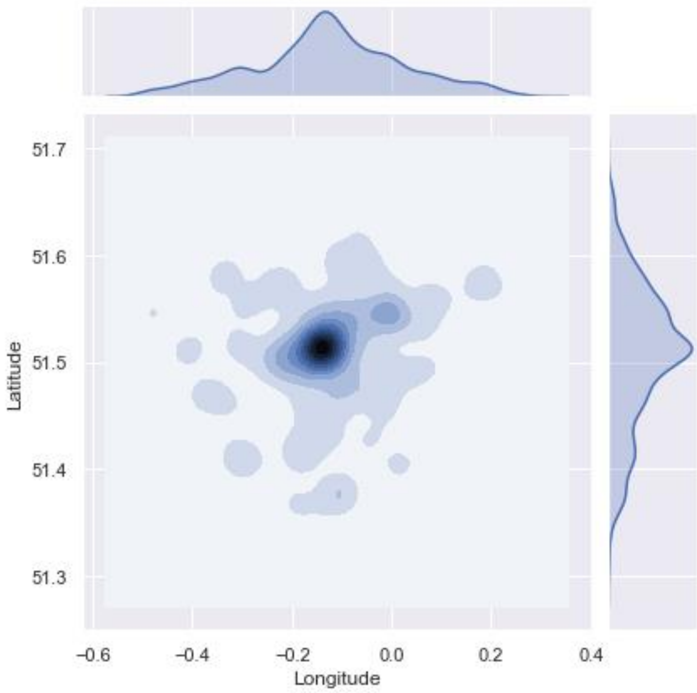


Figure 9.39: The estimated joint and marginal densities for shoplifting incidents in November 2018

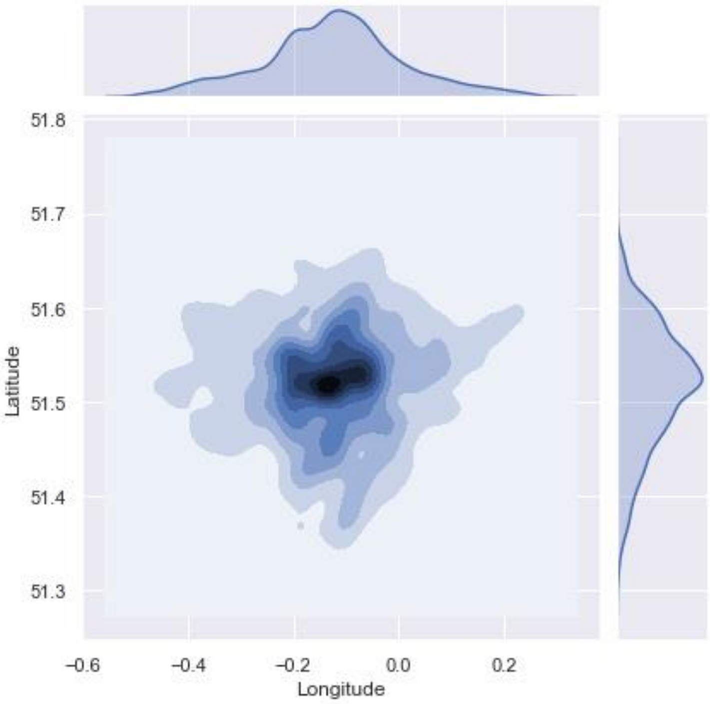


Figure 9.40: The estimated joint and marginal densities for burglaries in July 2018

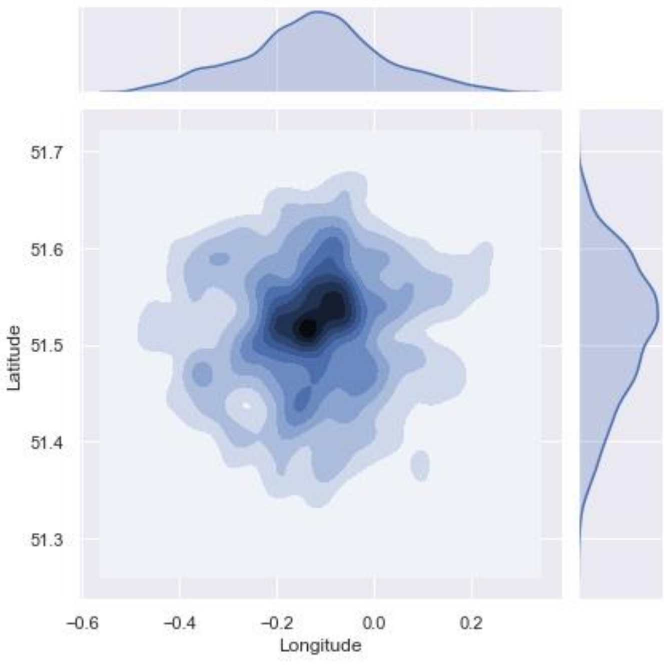


Figure 9.41: The estimated joint and marginal densities for burglaries in October 2018

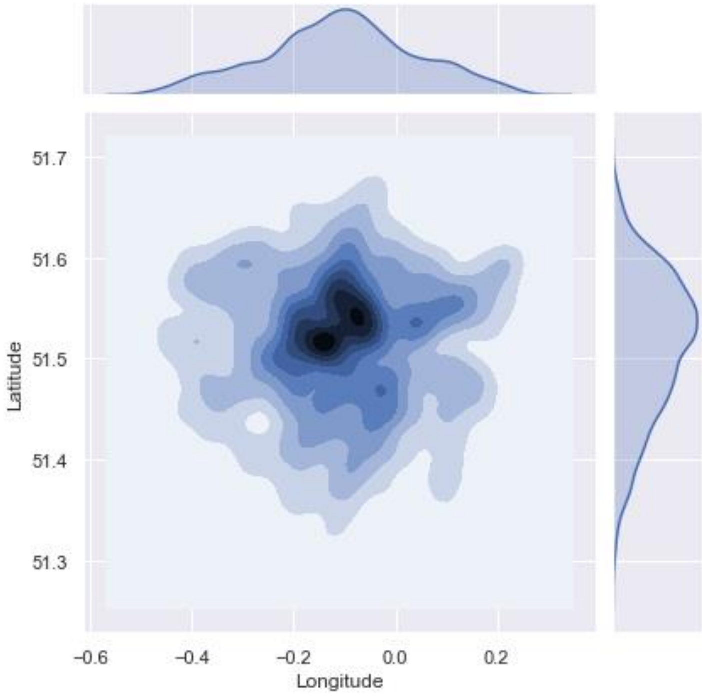


Figure 9.42: The estimated joint and marginal densities for burglaries in December 2018