Name = GAUTAM Kumar

roll =B19EE031

Lab 7 report

PART 1:

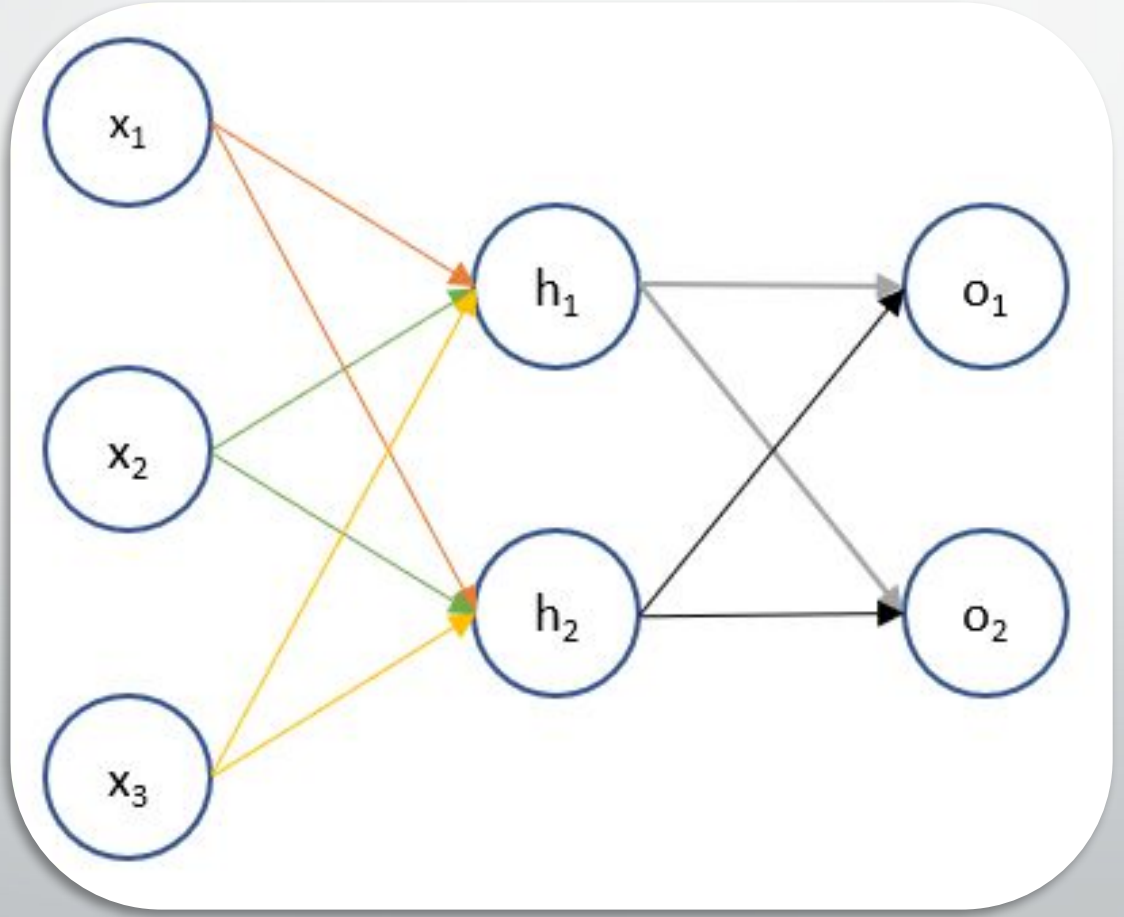
Creating a MLP from scratch

- *Creating a model with 3 inputs and 2 target variables. We have a hidden layer with 2 nodes.*
- *Both layers, the input layer and the hidden layer have a bias node.*
- *Sigmoid activation is used.*
- *Error function is mean squared error.*

Initializations

- *Weights :* `[0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.1]`
- *Biases :* `[0.5, 0.5]`
- *Inputs :* `[1, 4, 5]`
- *Target variables :* `[0.1, 0.05]`
- *Learning rate : 0.01*

Model

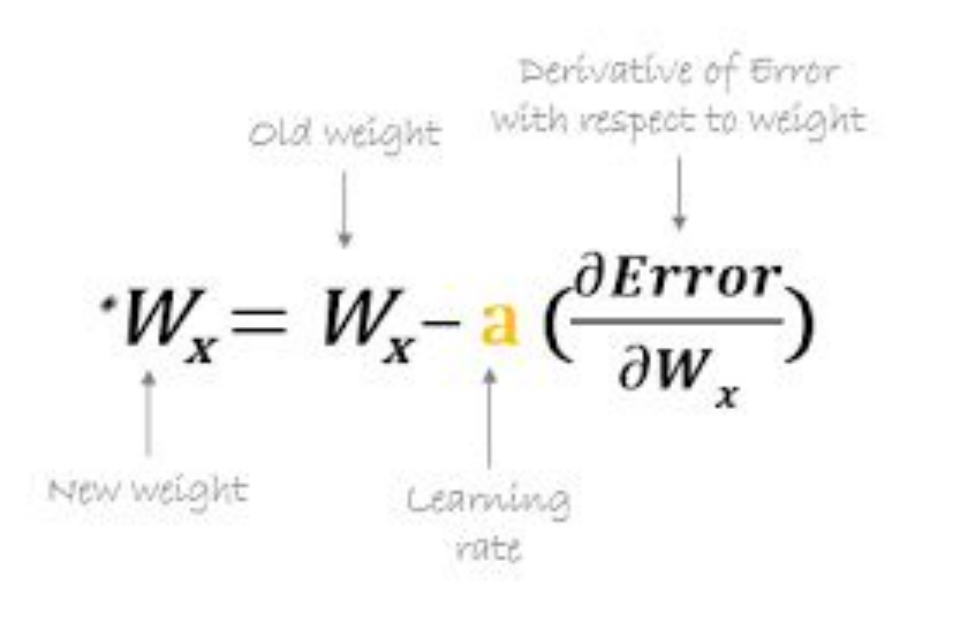


Steps involved in training

- *Initializing weights and biases.*
- *Forward propagation through the output values.*
- *Using proper error or loss function. (SSE in this case)*
- *Backpropagation and determination of error derivatives.*
- *Updating the values of weights and biases after each iteration.*

UPDATING PARAMETERS

Updating weights



Updating biases

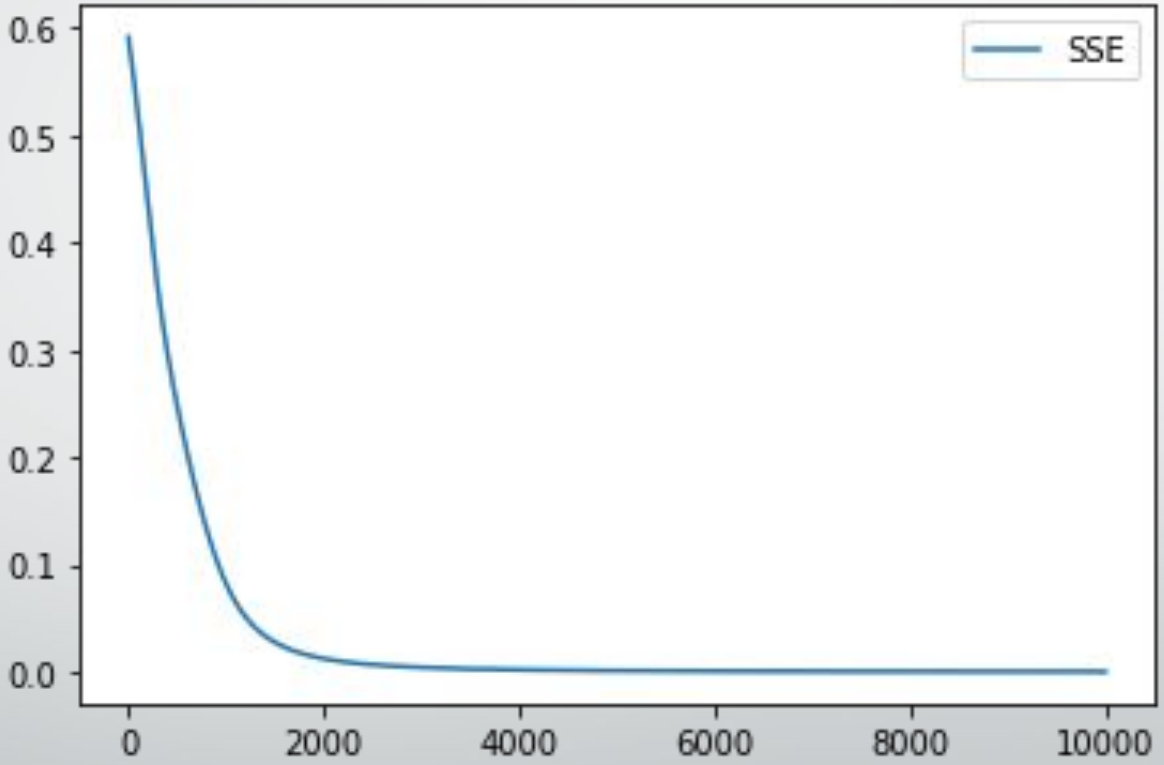
$$b_1 := b_1 - \alpha \frac{dE}{db_1}$$

$$b_2 := b_2 - \alpha \frac{dE}{db_2}$$

UPDATING ALL PARAMETERS

```
#Updating all parameters
w1 = w1 - alpha * dE_dw1
w2 = w2 - alpha * dE_dw2
w3 = w3 - alpha * dE_dw3
w4 = w4 - alpha * dE_dw4
w5 = w5 - alpha * dE_dw5
w6 = w6 - alpha * dE_dw6
w7 = w7 - alpha * dE_dw7
w8 = w8 - alpha * dE_dw8
w9 = w9 - alpha * dE_dw9
w10 = w10 - alpha * dE_dw10
b1 = b1 - alpha * dE_db1
b2 = b2 - alpha * dE_db2
wlist = [w1,w2,w3,w4,w5,w6,w7,w8,w9,w10]
blist = [b1,b2]
```

Variation of Error with each iteration



Extra Analysis: Using keras to build the same model

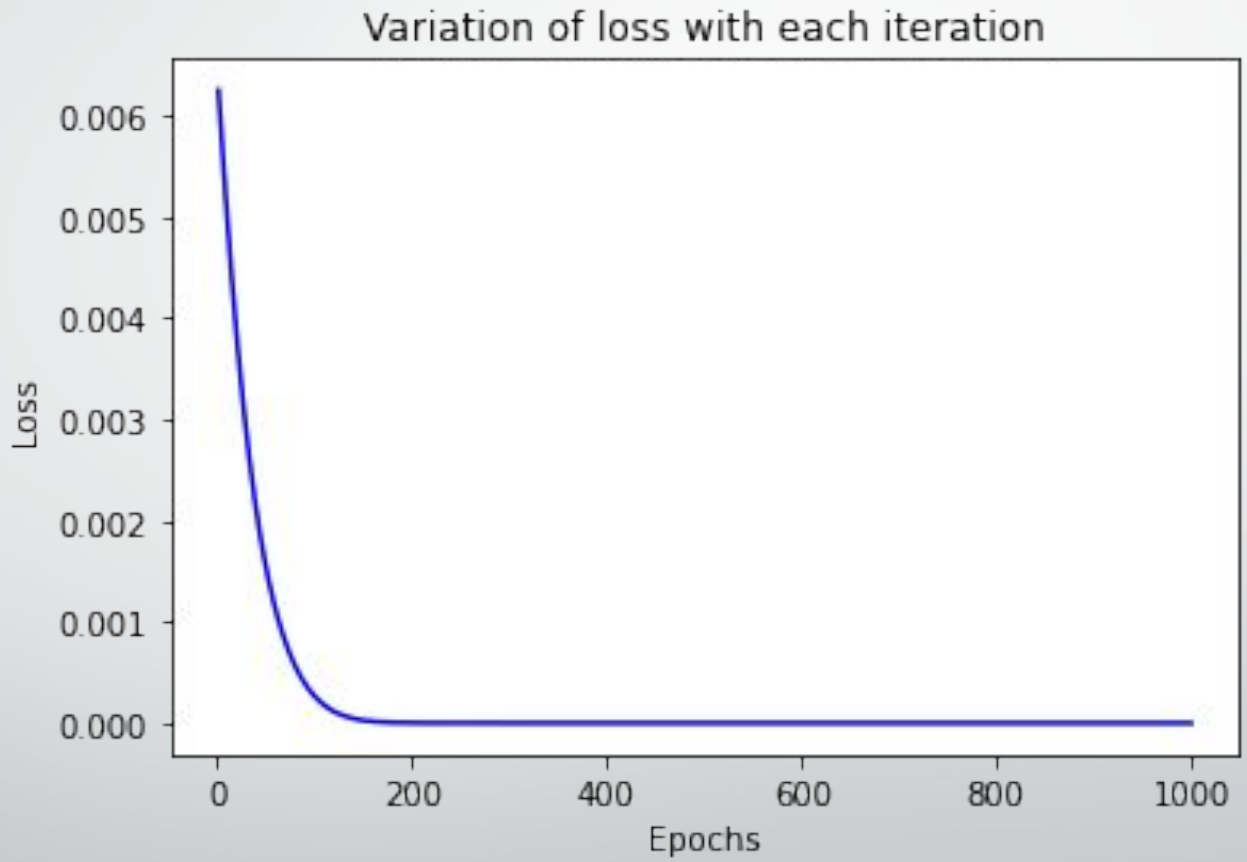
```
Epoch 1/1000
1/1 [==============================] - 1s 664ms/step - loss: 0.0063 - accuracy: 1.0000
Epoch 2/1000
1/1 [==============================] - 0s 6ms/step - loss: 0.0061 - accuracy: 1.0000
Epoch 3/1000
1/1 [==============================] - 0s 5ms/step - loss: 0.0060 - accuracy: 1.0000
Epoch 4/1000
1/1 [==============================] - 0s 6ms/step - loss: 0.0058 - accuracy: 1.0000
Epoch 5/1000
1/1 [==============================] - 0s 7ms/step - loss: 0.0057 - accuracy: 1.0000
Epoch 6/1000
1/1 [==============================] - 0s 4ms/step - loss: 0.0055 - accuracy: 1.0000
Epoch 7/1000
1/1 [==============================] - 0s 4ms/step - loss: 0.0054 - accuracy: 1.0000
Epoch 8/1000
1/1 [==============================] - 0s 5ms/step - loss: 0.0053 - accuracy: 1.0000
Epoch 9/1000
1/1 [==============================] - 0s 5ms/step - loss: 0.0051 - accuracy: 1.0000
Epoch 10/1000
1/1 [==============================] - 0s 5ms/step - loss: 0.0050 - accuracy: 1.0000
Epoch 11/1000
1/1 [==============================] - 0s 6ms/step - loss: 0.0049 - accuracy: 1.0000
Epoch 12/1000
1/1 [==============================] - 0s 5ms/step - loss: 0.0047 - accuracy: 1.0000
Epoch 13/1000
1/1 [==============================] - 0s 5ms/step - loss: 0.0046 - accuracy: 1.0000
Epoch 14/1000
1/1 [==============================] - 0s 5ms/step - loss: 0.0045 - accuracy: 1.0000
```

We can the loss function decreasing after each iteration.

So we can say that training error decreases with each epoch.

Accuracy is always 1 because our model is relatively very simple and has only one training example.

Variation of loss with epochs in keras model



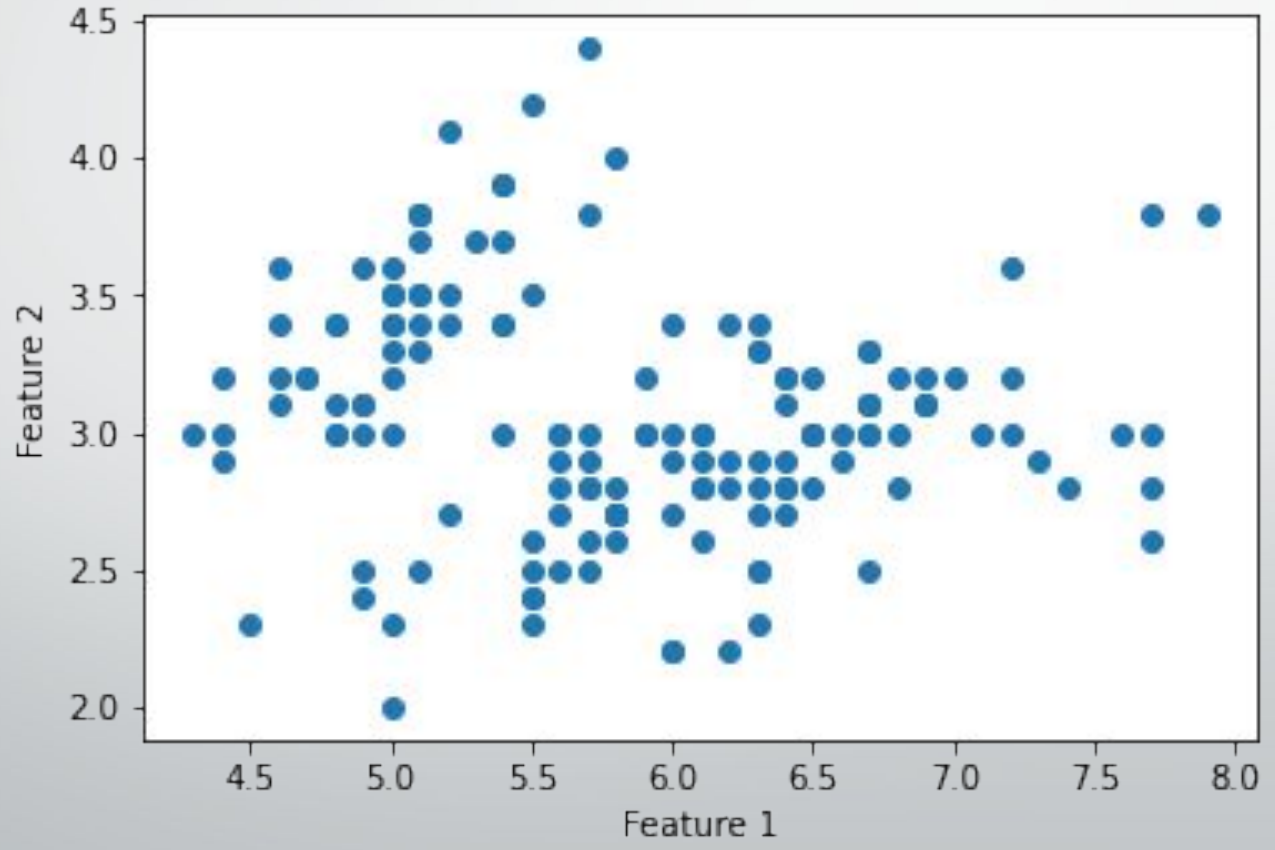
Analysis

- *Derivatives were calculated to update the parameters.*
- *Due to subsequent derivatives, weights of initial layers are not trained properly.*
- *W1 = W1 – alpha[d(Error)/dW2 * alpha* d(Error)/dW1]*
- *Here alpha is learning rate. Lower values of alpha can lead to problem of vanishing gradient and higher values can lead to improper training.*
- *Hence hyperparameters should be chosen properly.*

Part 2: KMeans model

Unsupervised Learning
Dataset : Iris
Number of clusters = 3

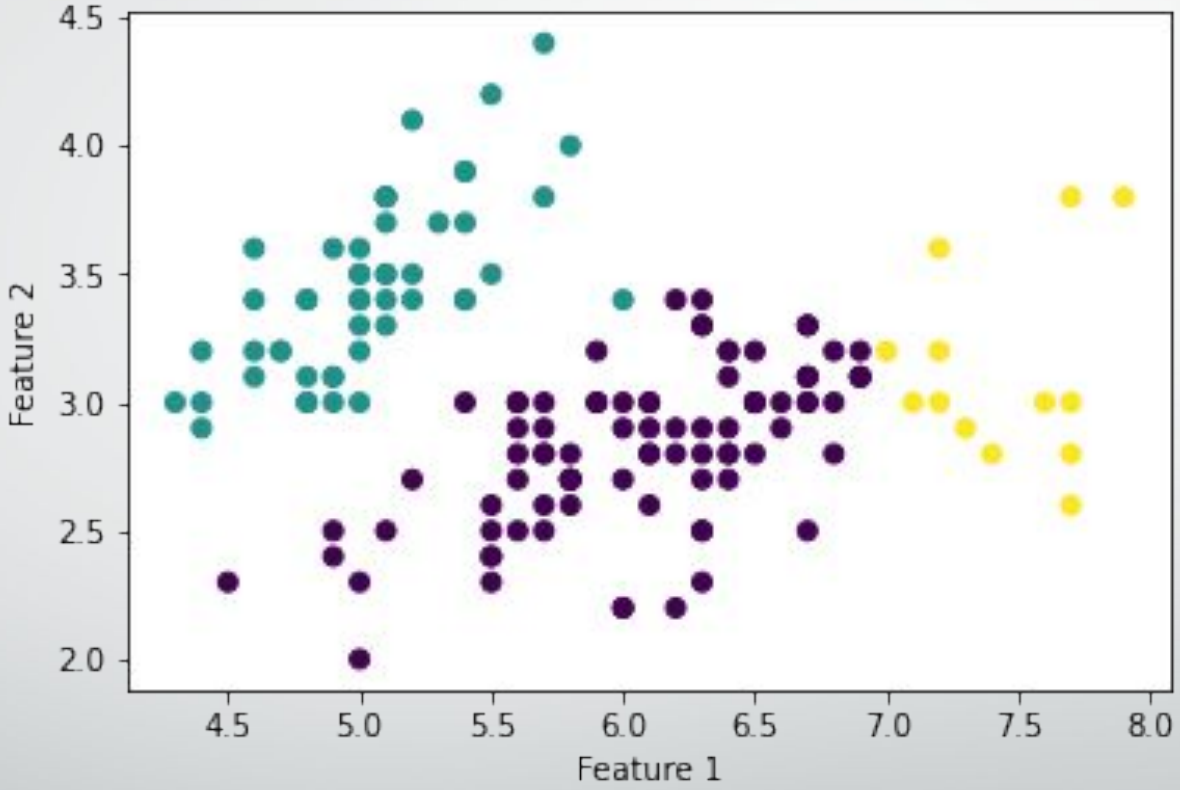
Variation of feature1 with feature2



Steps involved in kmeans clustering

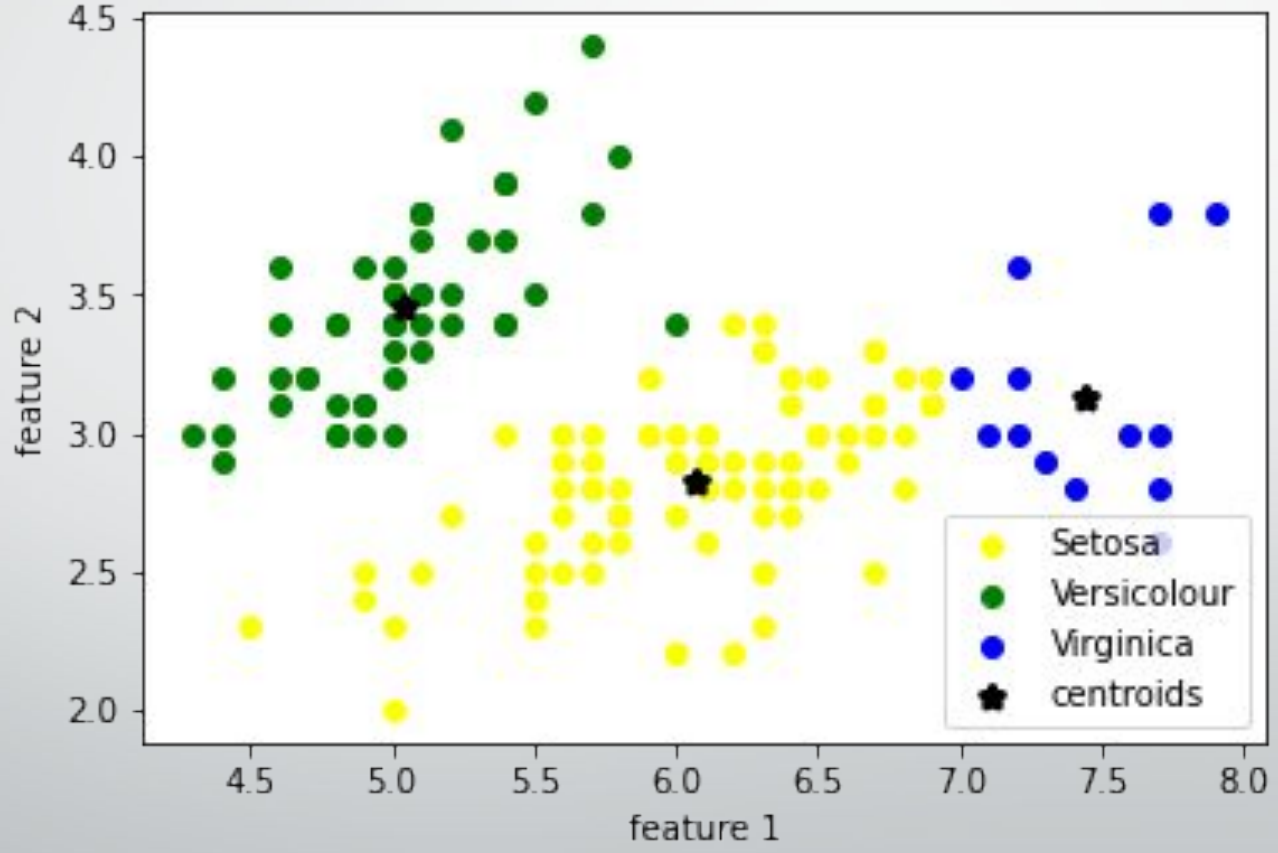
- *First we find the number k, means the number of clusters to be made.*
- *Then there is random initialization of centroids.*
- *Then clusters are created, meaning the points closer to one of the centroid is assigned that particular cluster.*
- *The centroids of each cluster is calculated again and there is redistribution of cluster points.*
- *The same process is repeated until we have stable cluster points or reach the end of maximum iteration.*

Clusters division

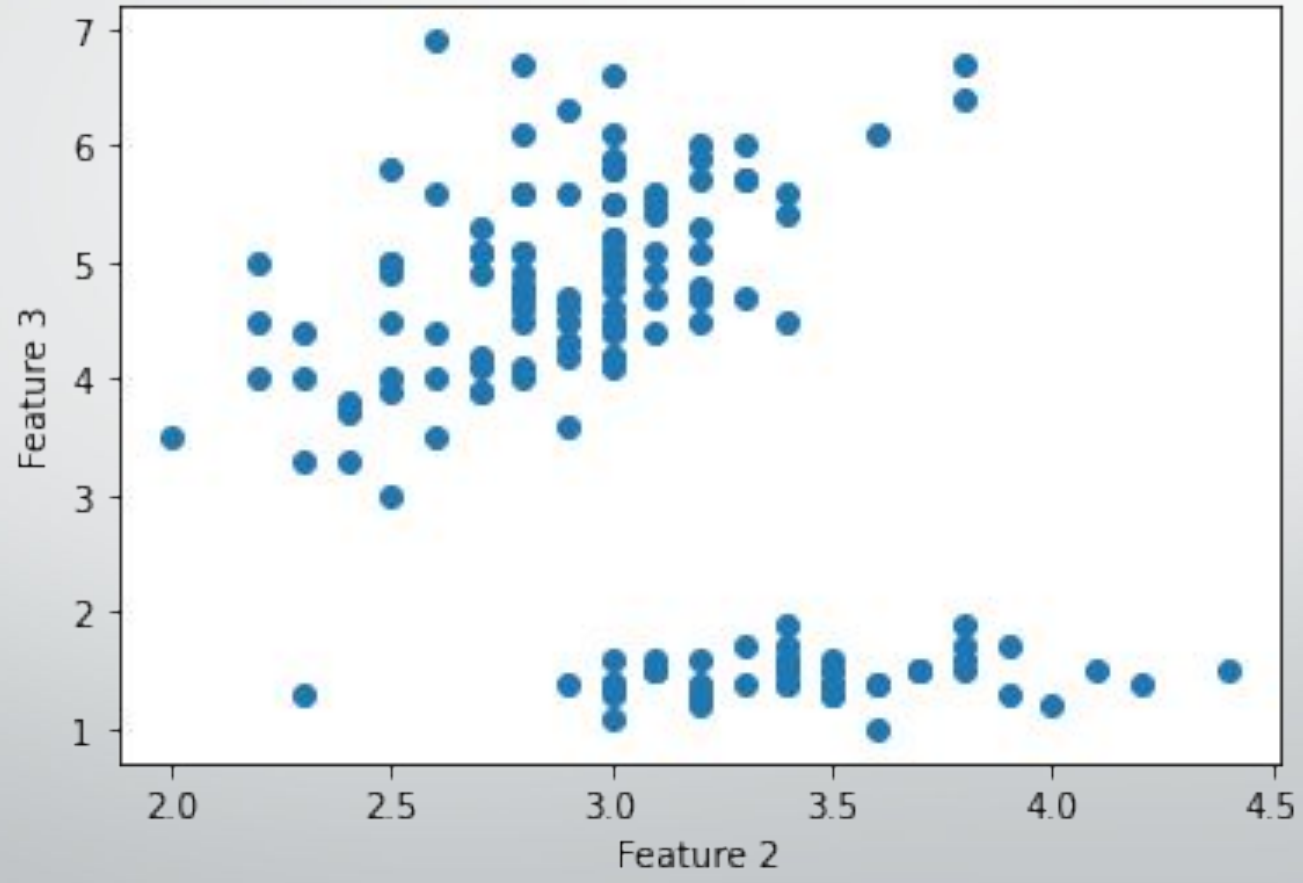


Redistribution stopped at 2nd iteration as there was no change in coordinates of centroids.

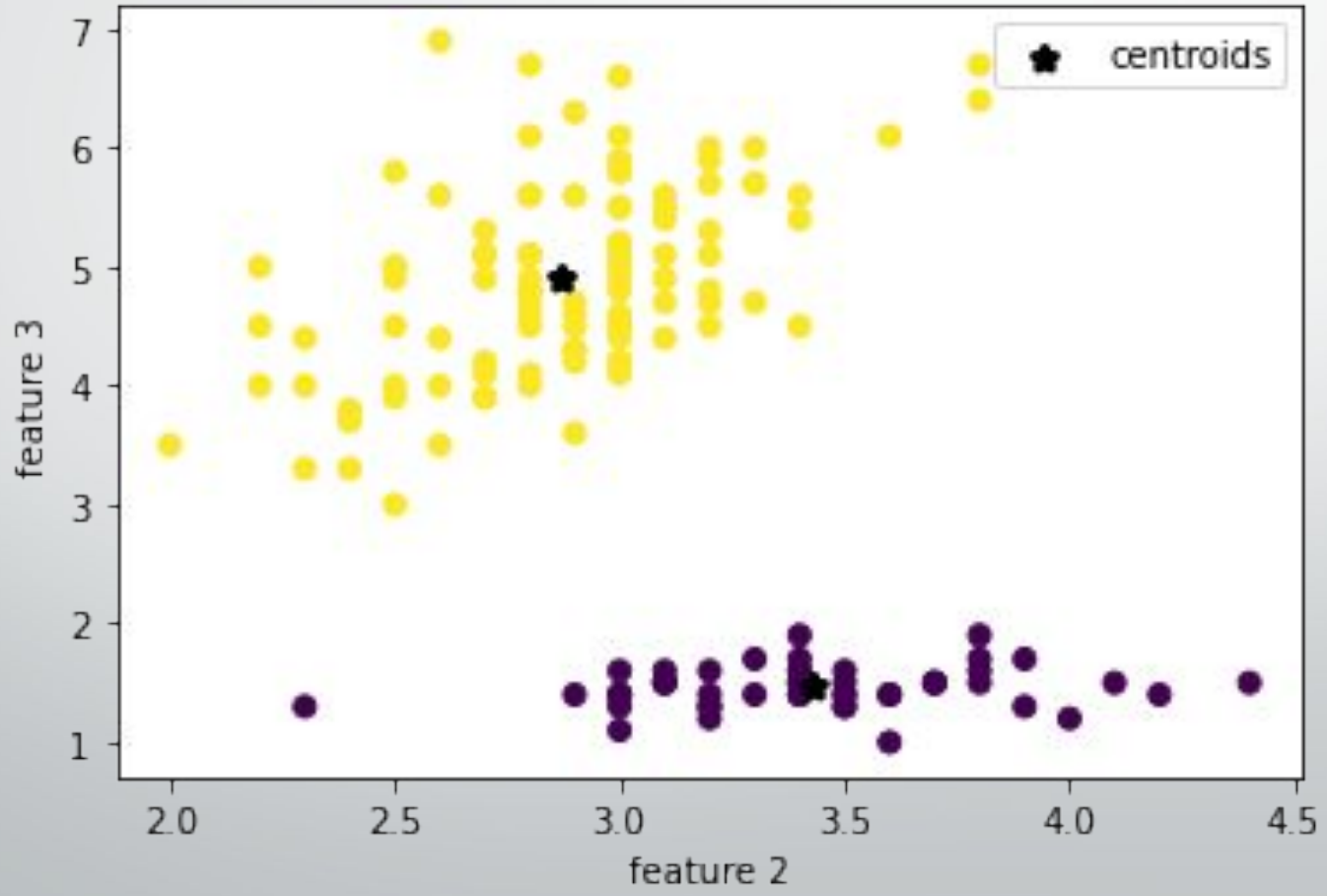
Clusters with centroids



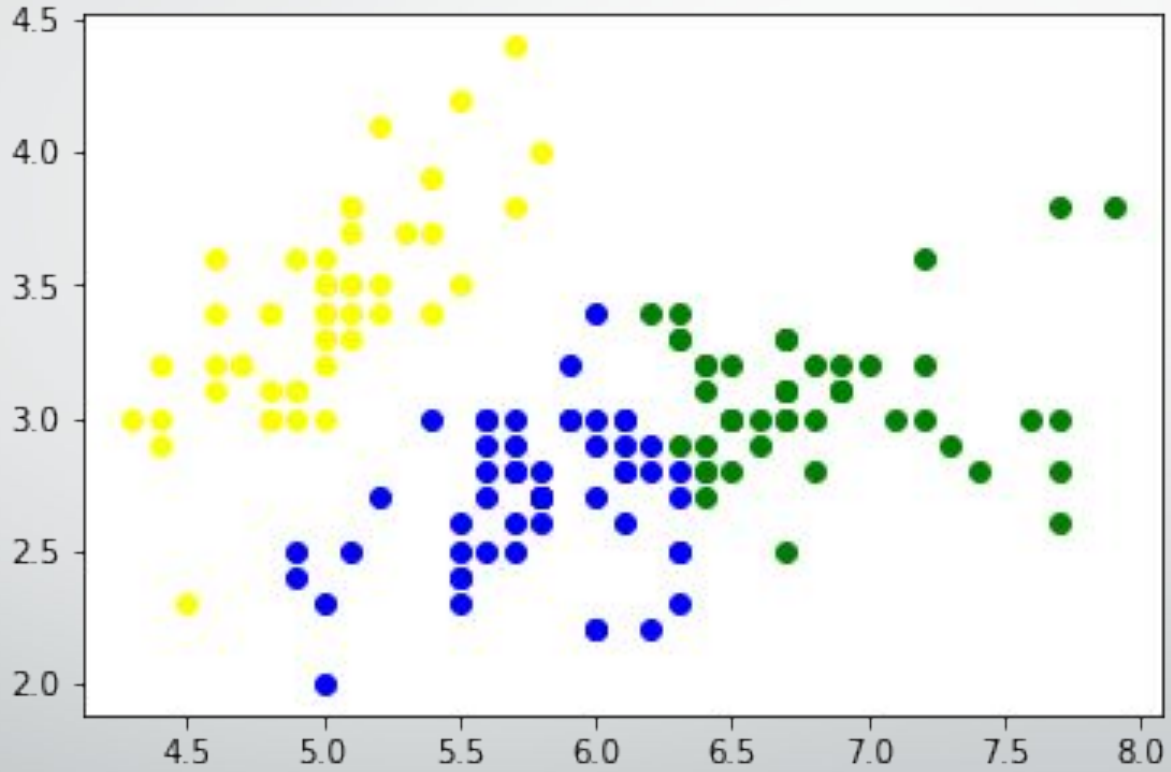
Distribution of feature 2 and feature 3



Clusters division with centroids

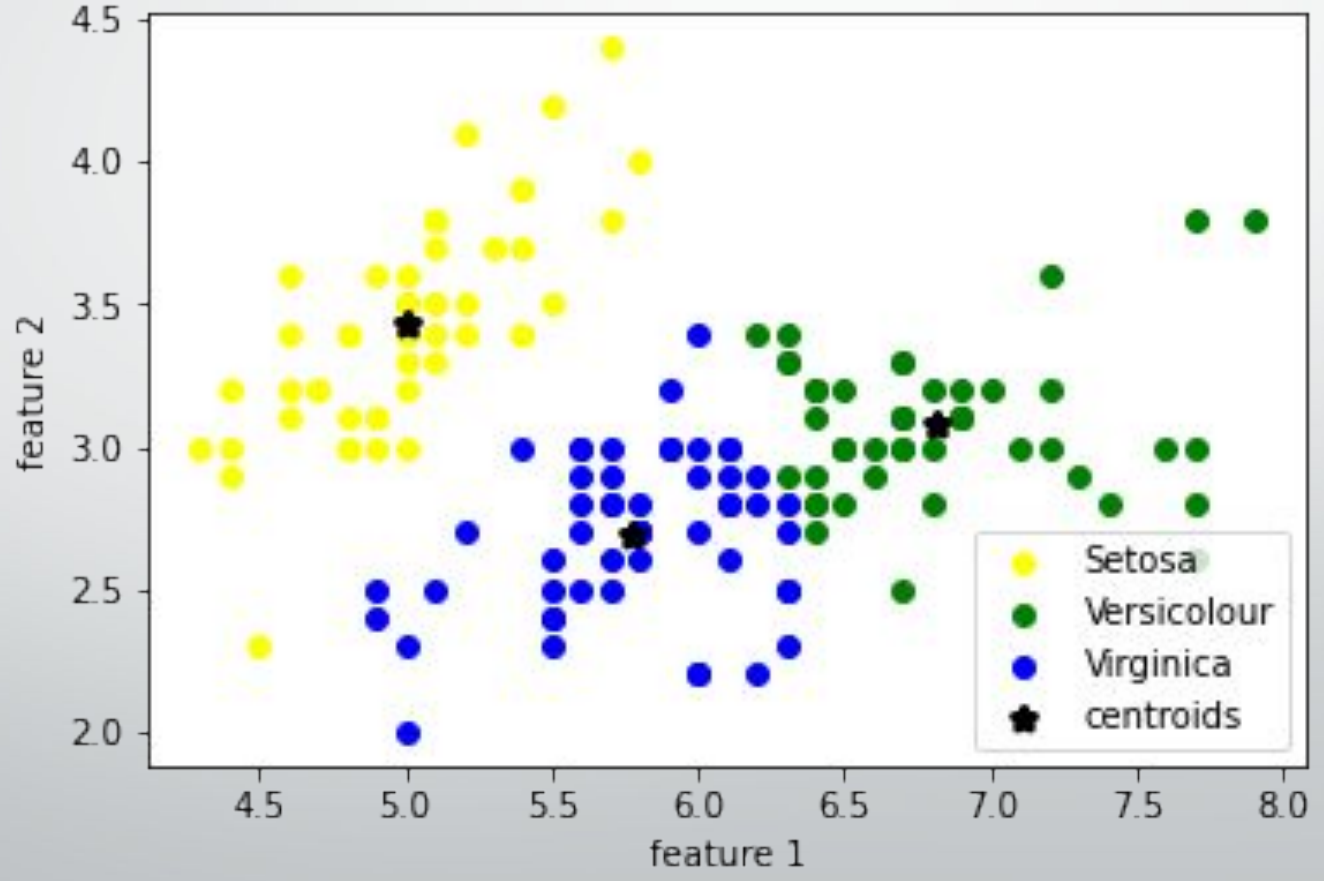


Cluster Distribution by inbuilt kmeans

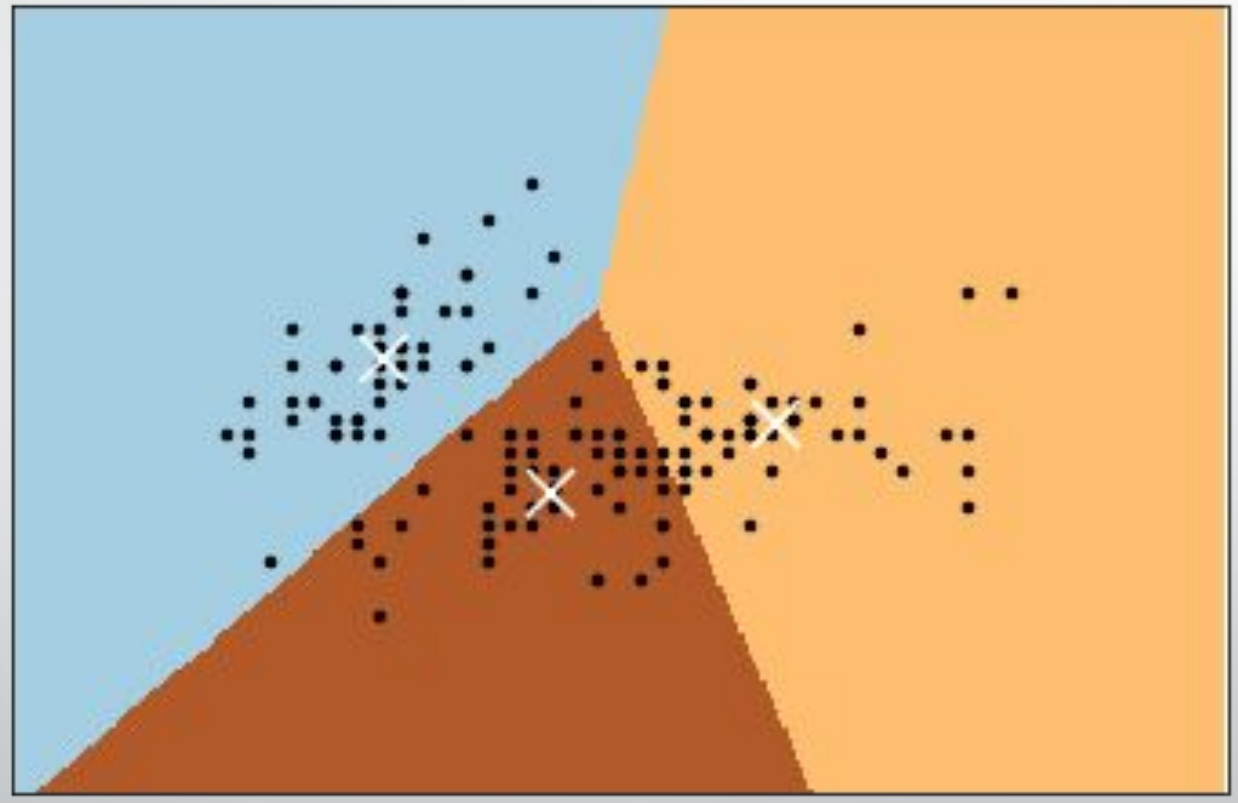


Sum of squared error: `37.050702127659578`

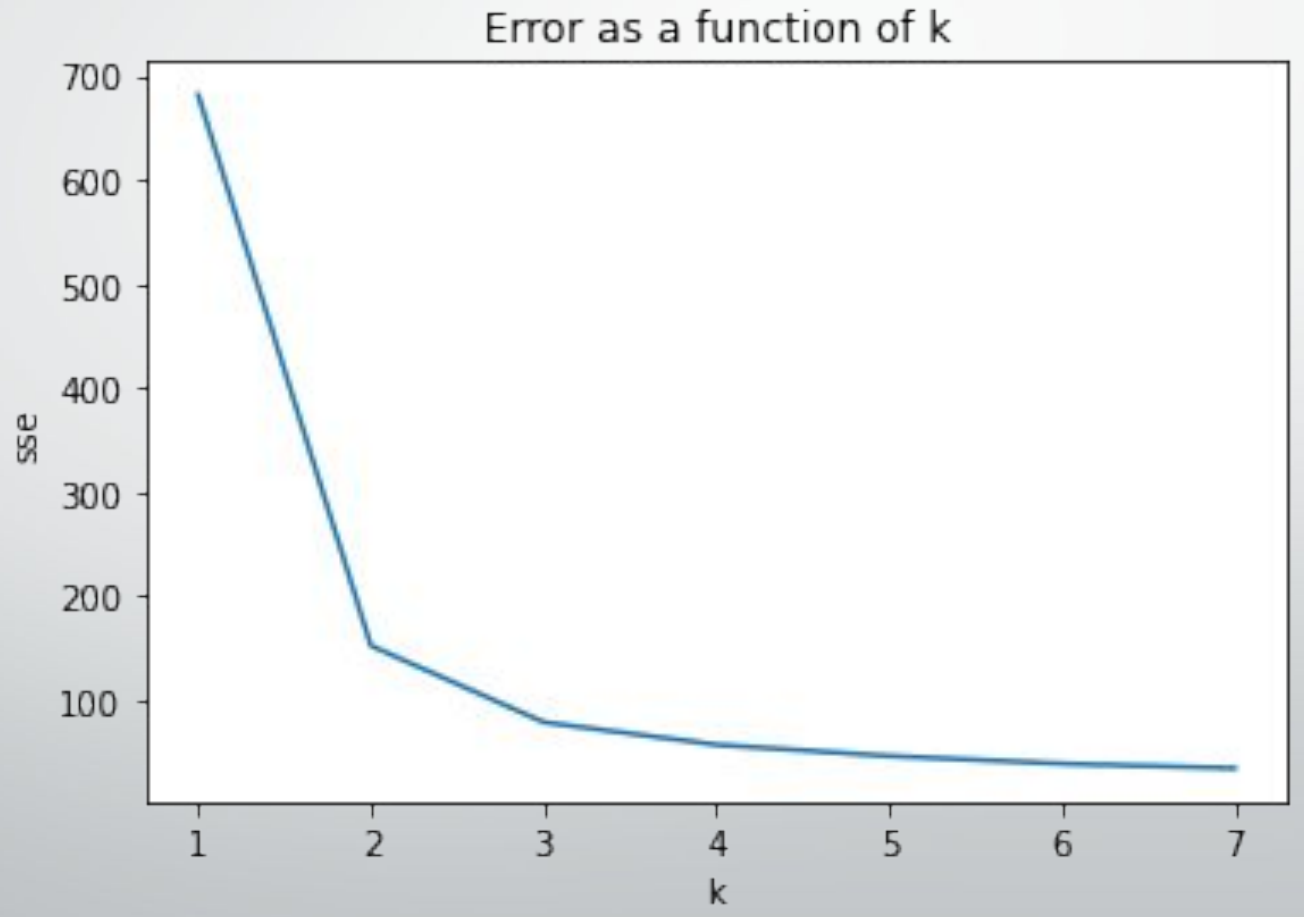
Clusters with centroids



Extra analysis: Clusters with decision boundaries



Extra Analysis: Variation of Error with k



Analysis

- *We can see that Sum of squared error decreases with increase in k, as the number of centroids increases.*
- *In unsupervised learning we don't have labels and we simply group the data points on the basis of clusters.*
- *Number of clusters, means the value of k can be found by the elbow method.*
- *As we can see in the graph of error as a function of k, the position of elbow vertex occurs at k=3, so that value is the most appropriate value to form k-clusters.*

Conclusion

- *Hence we learnt about MLP (Multi Layer Perceptron) model.*
- *We learnt the method to build a model from scratch.*
- *We also saw the working of keras model and analyzed the error as a function of epochs.*
- *We learnt about kmeans model.*
- *We found the decision boundary and calculated error as a function of k.*

We also saw how to choose appropriate value of 'k' for the model.