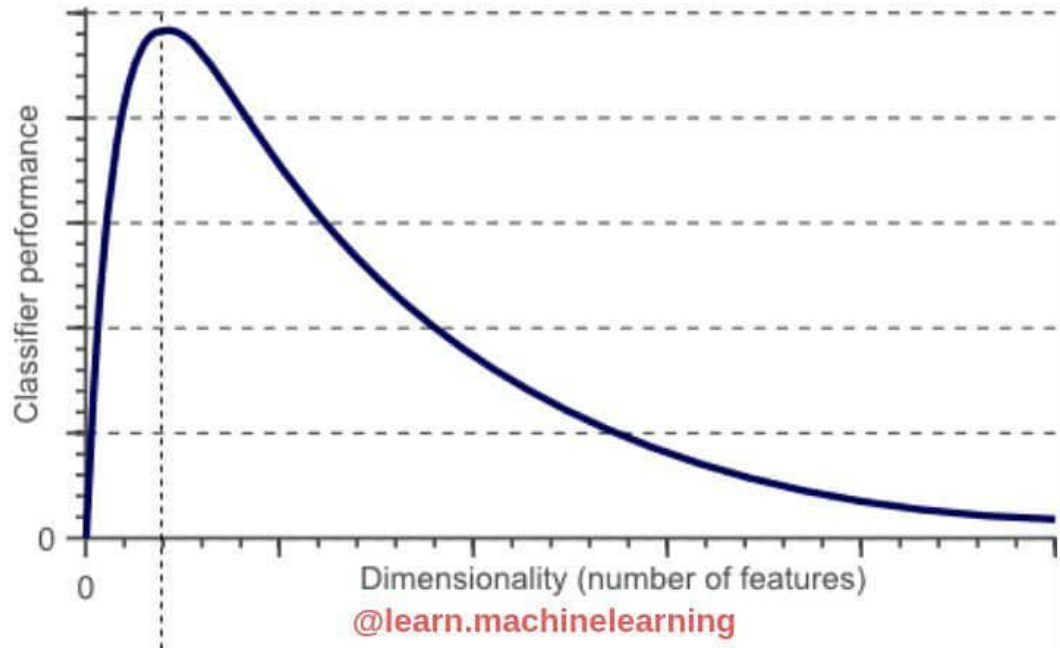
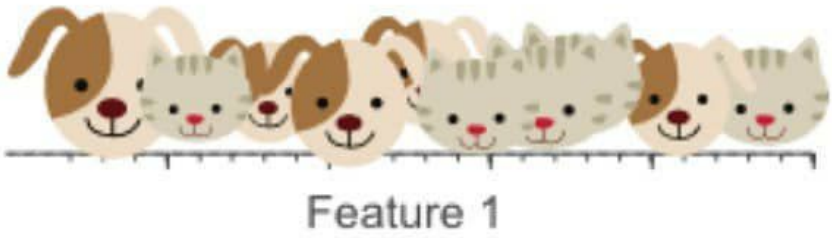
Curse of dimensionality

- As the number of features or dimensions grows, the amount of data we need to generalise accurately grows exponentially.
- If you have a lot of input dimensions, then your problem becomes computationally expensive and difficult to solve - True but Why?.
- we can say that by increasing number of features data density decreases and complexity increases and it became very difficult for machine learning model to work efficiently.

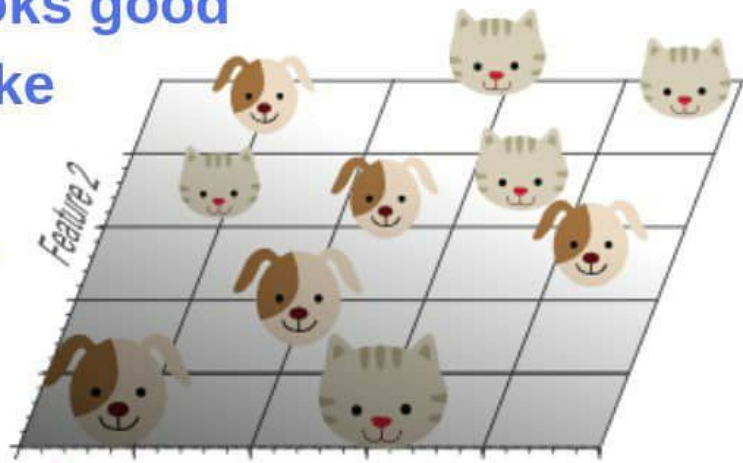


Curse of dimensionality

- Let's see why this happens with an example of Dog and Cat classifier.
- first, we solve the above problem using only one feature like height and we have only one dimension data and it is easy to represent on a line like this separation is not perfect. But still, it works better.

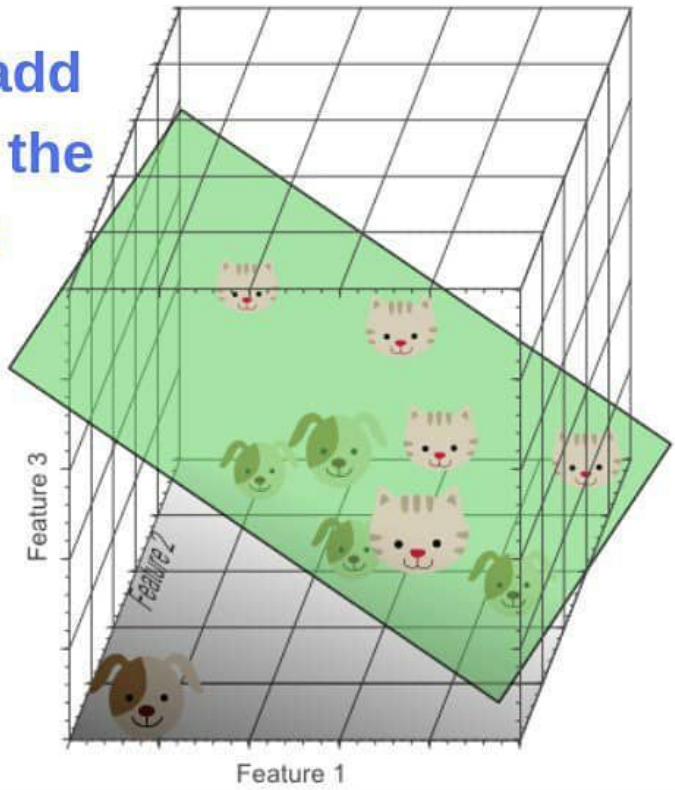


- So we thought by adding another feature can increase our accuracy so we added the paw size. Now we have 2D data. It looks good but still, we still cannot make a perfect linear separation between our observations.



Curse of dimensionality

- We can add another feature like weight and it looks like By adding the last feature our classifier is able to create a linear combination of the three features in order to obtain perfect classification results on our training dataset.
- By this, we would think to add more features will improve the classification accuracy but not.



Moreover, note that the density of our training samples decreased exponentially as we increased the dimensionality of the problem which also can become a problem.

Curse of dimensionality

- As we add more features, the space between the observations grows, and it becomes sparser and sparser.
- This sparsity helps our classifier to classify our observations since it becomes more easy to find a separable hyperplane due to the fact that the likelihood of our training samples lying on the wrong side of the best hyperplane becomes infinitely small as you increase the number of features to infinite. However, by projecting this high dimensional classification into a lower-dimensional space we are struck by an evident problem in Machine Learning: Overfitting!
- Then how can we find the best dimension of the data?

Curse of dimensionality

Avoiding curse of dimensionality

- There is no fixed rule that defines how many features should be used in a regression/classification problem.
- Mostly we use dimensionality reduction techniques to solve this.
- PCA
- t-SNE (non-parametric/ nonlinear)
- Sammon mapping (nonlinear)
- SNE (nonlinear)
- MVU (nonlinear)
- Laplacian Eigenmaps (nonlinear)