Multinomial Naive Bayes

Multinomial Naïve Bayes

- Multinomial Naïve Bayes is designed for text
  - based on word appearance only, not non-appearance
  - can account for multiple repetitions of a word
  - treats common words differently from unusual ones
- It's a lot faster than plain Naïve Bayes!
  - ignores words that do not appear in a document
  - internally, Weka uses a sparse representation of the data
- The StringToWordVector filter has many interesting options
  - although they don't necessarily give the results you're looking for!
  - outputs results in "sparse data" format, which MNB takes advantage of

Example- Normal Vs Spam mail

- We received a lot of emails from friends, family, office and we also receive spam mails. Initially, we consider eight normal messages and four spam messages.
- Let see the histogram of all the words that occur in the normal messages from family and friends.

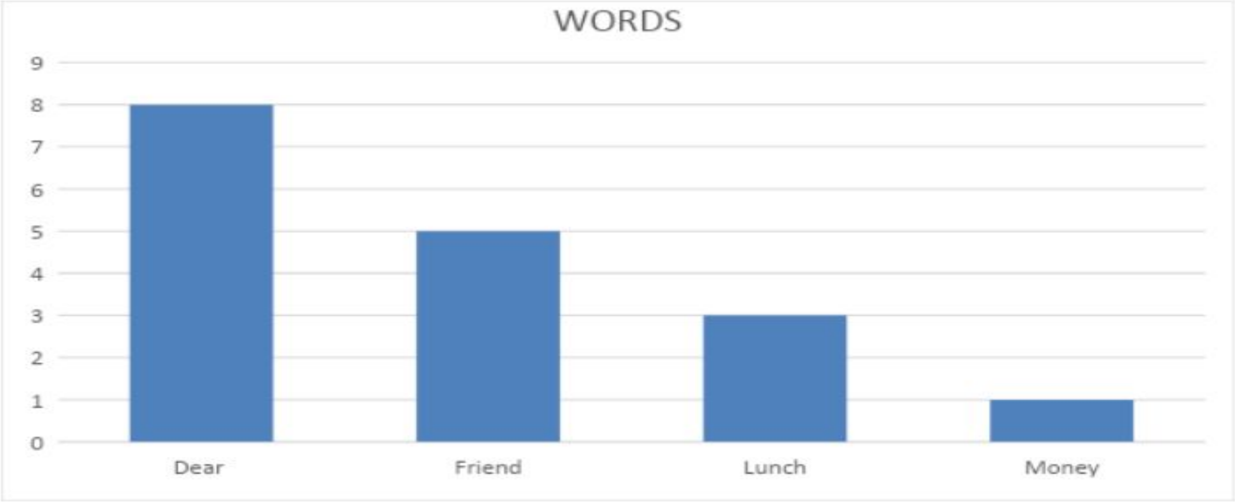
Normal mail

- We can use the histogram to calculate the probabilities of seeing each word, given that it was a normal message.
  The probability of word dear given that we saw in normal message is-

- **Probability (Dear|Normal) = 8 /17 = 0.47**

Similarly, the probability of word Friend is-

- Probability (Friend/Normal) = 5/ 17 =0.29
- Probability (Lunch/Normal) = 3/ 17 =0.18
- Probability (Money/Normal) = 1/ 17 =0.06



Normal Vs Spam mail

Now, let’s say we have received a normal message as **Dear Friend** and we want to find out if it’s a normal message or spam.

- We start with an initial guess that any message is a Normal Message.
- From our initial assumptions of 8 Normal messages and 4 Spam messages, 8 out of 12 messages are normal messages. The prior probability, in this case, will be:
- **Probability (Normal) = 8 / (8+4) = 0.67**
- We multiply this prior probability with the probabilities of **Dear Friend** that we have calculated earlier.
- **0.67 * 0.47 * 0.29 = 0.09**

0.09 is the probability score considering **Dear Friend** is a normal message.

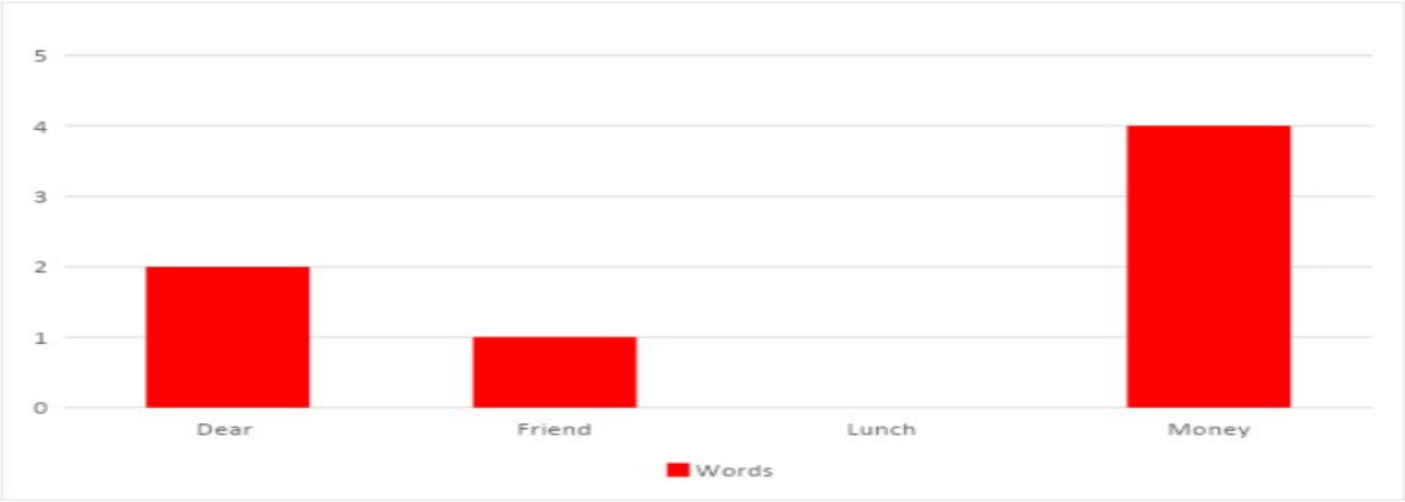
Spam mail

- The probability of word dear given that we saw in spam message is-

  Probability (Dear|Spam) = 2 /7 = 0.29

- Similarly, the probability of word Friend is-

  Probability (Friend/Spam) = 1/ 7 =0.14
  Probability (Lunch/Spam) = 0/ 7 =0.00
  Probability (Money/Spam) = 4/ 7 =0.57



Normal Vs Spam mail

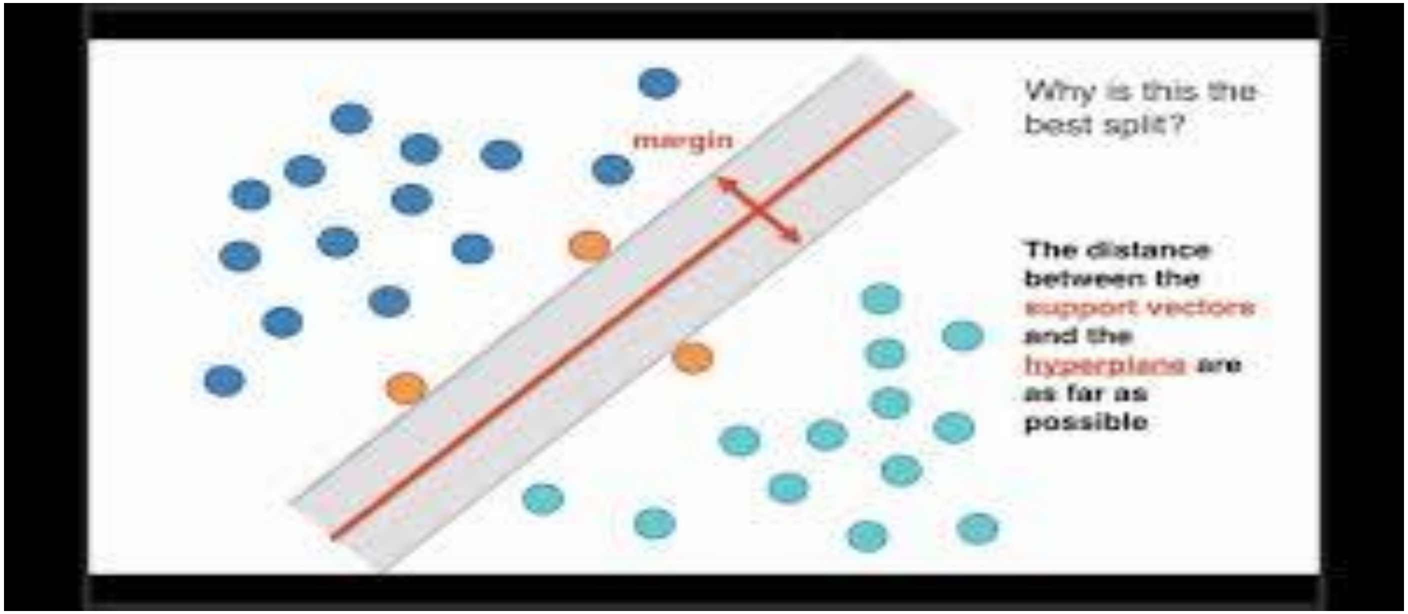
- Alternatively, let’s say that any message is a Spam.
  - 4 out of 12 messages are Spam. The prior probability in this case will be:
  - **Probability (Normal) = 4 / (8+4) = 0.33**
  - Now we multiply the prior probability with the probabilities of **Dear Friend** that we have calculated earlier.
  - **0.33 * 0.29 * 0.14 = 0.01**

0.01 is the probability score considering **Dear Friend** is a Spam.

Conclusion- The probability score of **Dear Friend** being a normal message is greater than the probability score of **Dear Friend** being spam. We can conclude that **Dear Friend** is a normal message.

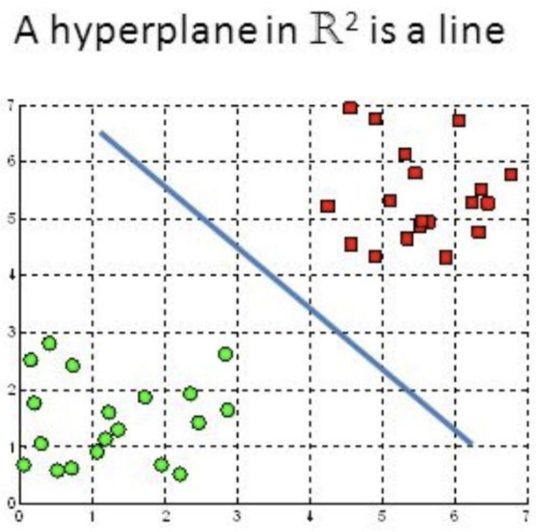
What is a Support Vector Machine?

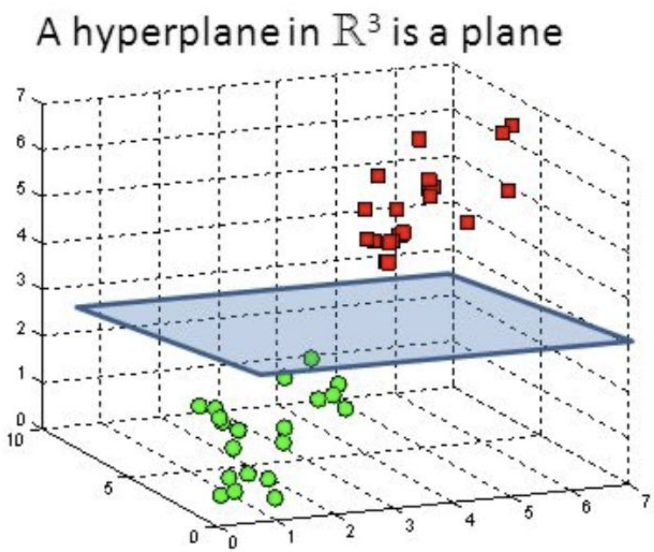
- **Support Vector Machine** is a discriminative algorithm that tries to find the optimal hyperplane.
- In a 2D space, a hyperplane is a line that optimally divides the data points into two different classes.
- In a higher-dimensional space, the hyperplane would have a different shape rather than a line.



Support Vector Machine

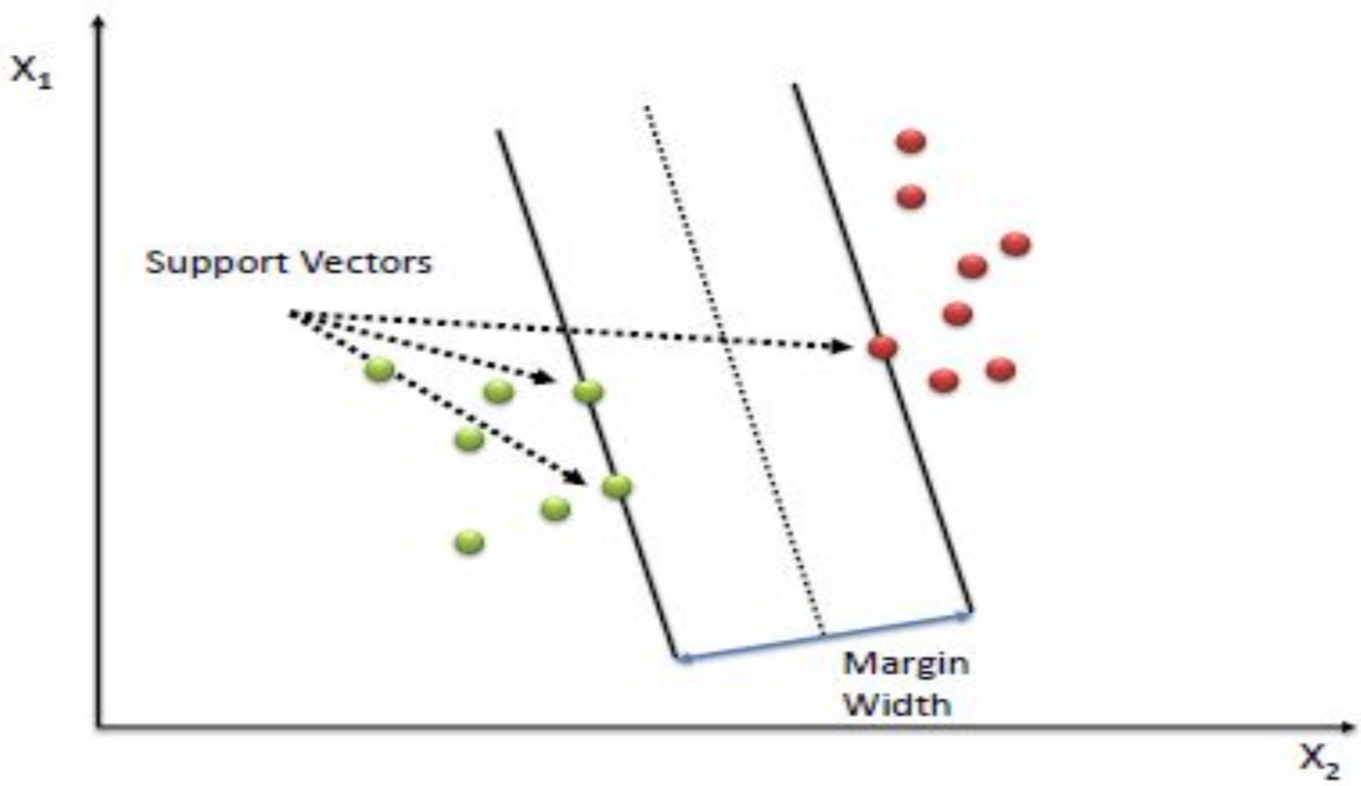
- **Hyperplane**: A hyperplane is a plane which is used to divide categories based on their values.
- A hyperplane is always 1 dimension less than the actual plane used for plotting the outcomes or for analyses.
  - Linear Regression with 1 feature and 1 outcome we can make a 2-D plane to depict the relationship and the regression line fitted to that is a 1-D plane.
  - Similarly, for a 3-D relationship, we get a 2-D hyperplane.





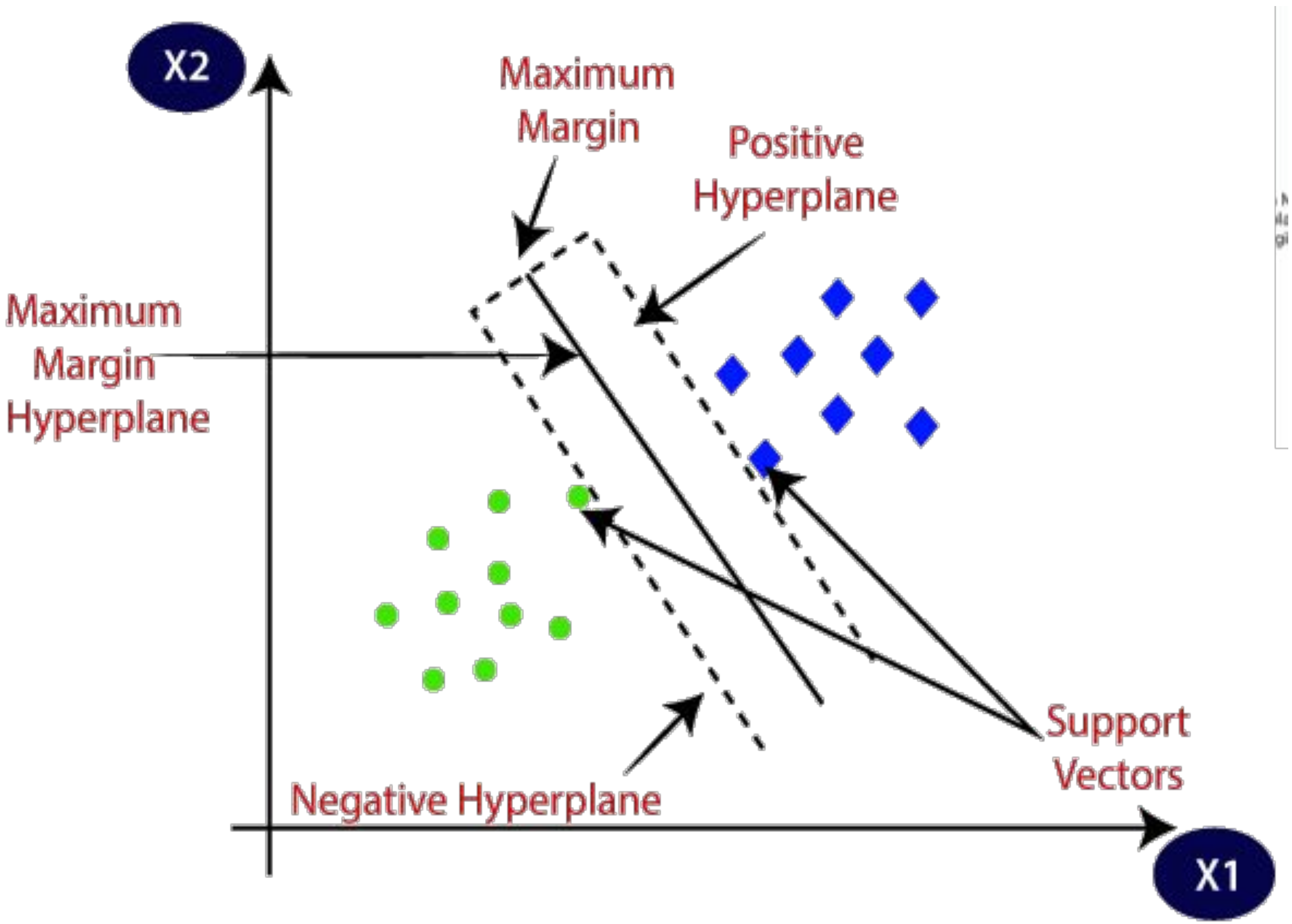
Support Vector Machine

- **Support Vectors**: Support Vectors are those points in the space that are closer to the hyperplane and also decide the orientation of the hyperplane.
- The lines or planes drawn is called Support Vector Lines or Support Vector Planes.



Margin Width

The perpendicular distance between the 2 support vector lines or planes is called Margin Width.

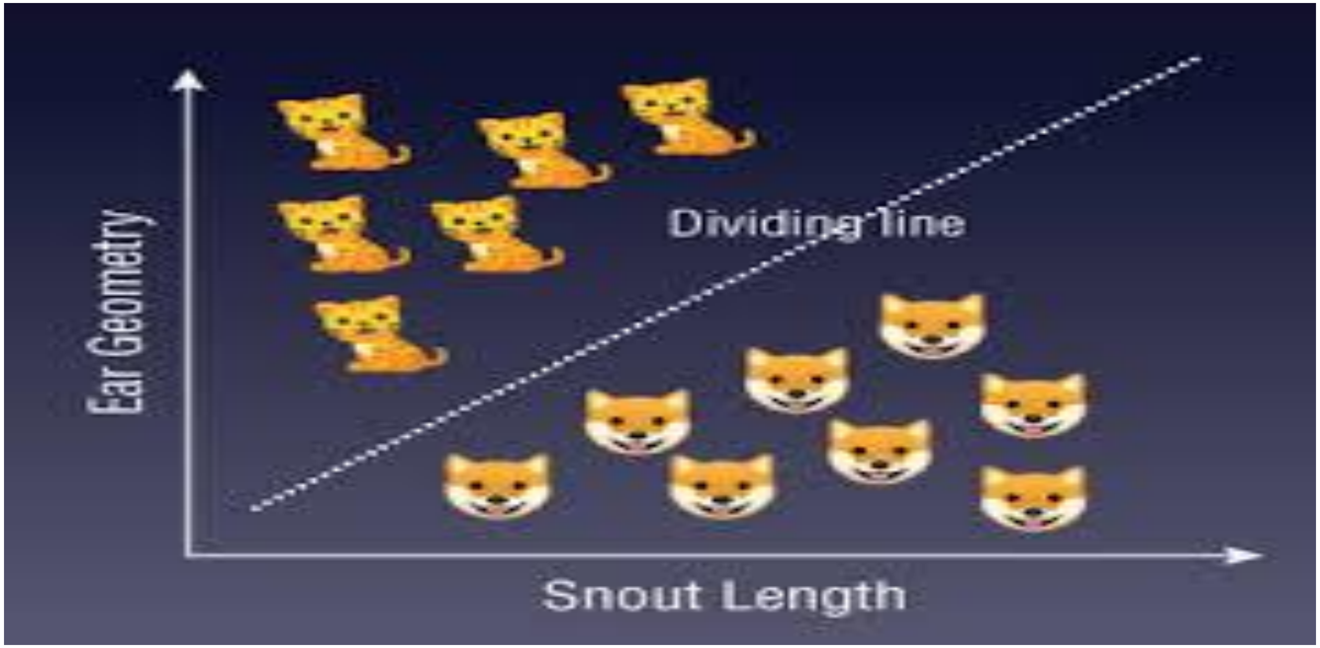


SVM example

Suppose we see a strange cat that also has some features of dogs, so if we want a model that can accurately identify whether it is a cat or dog

- We will first train our model with lots of images of cats and dogs so that it can learn about different features
- and then we test it with this strange creature.

- So as support vector creates a decision boundary between these two data and choose extreme cases (support vectors).



Support Vector Regression

The key advantages

- SVM works really well with high dimensional data. If your data is in higher dimensions, it is wise to use SVR.
- For data with a clear margin of separations, SVM works relatively well.
- When data has more features than the number of observations, SVM is one of the best algorithms to use.
- As a discriminative model, it need not memorize anything about data. Therefore, it is memory efficient.

Some drawbacks

- It is a bad option when the data has no clear margin of separation i.e. the target class contains overlapping data points.
- It does not work well with large data sets.
- For being a discriminative model, it separates the data points below and above a hyperplane. So, you will not get any probabilistic explanation of the output.
- It is hard to understand and interpret SVM as its underlying structure is quite complex.

INTERNSHIPSTUDIO

QUIZ!

Q.1 Explain Multinomial Naïve Bayes?

Q.2 Write the probabilities of different normal words coming in Multinomial Naïve Bayes example?

Q.3 Define support boundaries and margin in SVM?

Q.4 Define support vectors and hyperplane.

Q.5 What are the key advantages/disadvantages of SVR.

INTERNSHIPSTUDIO