

Chap4#5#2: Machine Learning for Anomaly-based Spam Detection

April 27, 2023



भारतीय प्रौद्योगिकी
संस्थान जम्मू
INDIAN INSTITUTE OF
TECHNOLOGY JAMMU

Devesh C Jinwala,
Professor, SVNIT and Adjunct Prof., CSE, IIT Jammu

Department of Computer Science and Engineering,
Sardar Vallabhbhai National Institute of Technology, SURAT

Topics to study in Chapter 4

- Machine learning for Anomaly Detection: Definition of an anomaly. Types of Anomalies or outliers in machine learning. Motivation for machine learning for anomaly detection.
Data Visualization. Supervised, Unsupervised and Semi-supervised Learning methods for Anomaly Detection.
Applications of Anomaly Detection: Intrusion detection, Fraud detection, Health monitoring, Defect detection, and lastly **Spam detection**. Intrusion Detection with Heuristics. Goodness-of-fit. Host Intrusion Detection. Network Intrusion Detection. Web Application Intrusion Detection.
Overview of Machine learning Approaches for Anomaly Detection:
Distance-based, Clustering-based and Model-based Approaches. Algorithms for Distance and Density-based approaches, Rank-based approaches, Ensemble Methods Algorithms for Time Series Data. Deep Learning for Anomaly Detection. Behavioural-based Anomaly Detection

[8 hours]

Topics in Handouts#1, #2, #3

1. Email Basics, Spam Detection Basics, and ML-based Spam Detection Basics
2. Classical ML-based Spam Filtering
3. Deep Learning-based Spam Filtering

ML-based Methods used for Email Spam filtering

ML-based Spam filtering: Clustering

Clustering-based ML Spam filtering. Clustering....

- is a type of approach used in dividing objects or case examinations into comparatively similar collections known as clusters.

¹J.S. Whissell, C.L.A. Clarke, Clustering for semi-supervised spam filtering, In: Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS '11), 2011, pp. 125–134.

²Spam e-mails filtering techniques. In: Int. J. Tech. Res. Appl., 4 (6), pp. 7 - 11, 2016

ML-based Spam filtering: Clustering

Clustering-based ML Spam filtering. Clustering....

- is a type of approach used in dividing objects or case examinations into comparatively similar collections known as clusters.
- algorithms are unsupervised learning tools are used on e-mail Spam datasets which usually have true labels.

¹J.S. Whissell, C.L.A. Clarke, Clustering for semi-supervised spam filtering, In: Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS '11), 2011, pp. 125–134.

²Spam e-mails filtering techniques. In: Int. J. Tech. Res. Appl., 4 (6), pp. 7 - 11, 2016

ML-based Spam filtering: Clustering

Clustering-based ML Spam filtering. Clustering....

- is a type of approach used in dividing objects or case examinations into comparatively similar collections known as clusters.
- algorithms are unsupervised learning tools are used on e-mail Spam datasets which usually have true labels.
- in research a good number of clustering algorithms have been shown to be effective i.e. to classify e-mail Spam datasets into either ham or Spam clusters (e.g. in Whissell and Clarke in¹),

¹J.S. Whissell, C.L.A. Clarke, Clustering for semi-supervised spam filtering, In: Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS '11), 2011, pp. 125–134.

²Spam e-mails filtering techniques. In: Int. J. Tech. Res. Appl., 4 (6), pp. 7 - 11, 2016

ML-based Spam filtering: Clustering

Clustering-based ML Spam filtering. Clustering....

- is a type of approach used in dividing objects or case examinations into comparatively similar collections known as clusters.
- algorithms are unsupervised learning tools are used on e-mail Spam datasets which usually have true labels.
- in research a good number of clustering algorithms have been shown to be effective i.e. to classify e-mail Spam datasets into either ham or Spam clusters (e.g. in Whissell and Clarke in¹),
- Two types of clustering methods that have been used for Spam classification as such.

¹J.S. Whissell, C.L.A. Clarke, Clustering for semi-supervised spam filtering, In: Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS '11), 2011, pp. 125–134.

²Spam e-mails filtering techniques. In: Int. J. Tech. Res. Appl., 4 (6), pp. 7 - 11, 2016

ML-based Spam filtering: Clustering

Clustering-based ML Spam filtering. Clustering....

- is a type of approach used in dividing objects or case examinations into comparatively similar collections known as clusters.
- algorithms are unsupervised learning tools are used on e-mail Spam datasets which usually have true labels.
- in research a good number of clustering algorithms have been shown to be effective i.e. to classify e-mail Spam datasets into either ham or Spam clusters (e.g. in Whissell and Clarke in¹),
- Two types of clustering methods that have been used for Spam classification as such.
- Density-based clustering and K-nearest neighbours (kNN).

¹J.S. Whissell, C.L.A. Clarke, Clustering for semi-supervised spam filtering, In: Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS '11), 2011, pp. 125–134.

²Spam e-mails filtering techniques. In: Int. J. Tech. Res. Appl., 4 (6), pp. 7 - 11, 2016

ML-based Spam filtering: Clustering

Clustering-based ML Spam filtering. Clustering....

- is a type of approach used in dividing objects or case examinations into comparatively similar collections known as clusters.
- algorithms are unsupervised learning tools are used on e-mail Spam datasets which usually have true labels.
- in research a good number of clustering algorithms have been shown to be effective i.e. to classify e-mail Spam datasets into either ham or Spam clusters (e.g. in Whissell and Clarke in¹),
- Two types of clustering methods that have been used for Spam classification as such.
- Density-based clustering and K-nearest neighbours (kNN).
 - **density based clustering** implemented in² showed the capacity to process encrypted messages too, thereby upholding privacy confidentiality.

¹J.S. Whissell, C.L.A. Clarke, Clustering for semi-supervised spam filtering, In: Proceedings of the 8th Annual Collaboration, Electronic Messaging, Anti-abuse and Spam Conference (CEAS '11), 2011, pp. 125–134.

²Spam e-mails filtering techniques. In: Int. J. Tech. Res. Appl., 4 (6), pp. 7 - 11, 2016

Clustering-based ML Spam filtering. Clustering....

- kNN proposed in³ is a distribution free method, - i.e. the data is not required to be drawn from a given probability distribution.

³

<https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbors-algorithm/>

Clustering-based ML Spam filtering. Clustering....

- kNN proposed in³ is a distribution free method, - i.e. the data is not required to be drawn from a given probability distribution.
 - this property is very vital as in the actual scenario, nearly all of the applied data disobey the standard hypothetical postulations made (such as Gaussian mixture, linearly separable, and others).

³

<https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbors-algorithm/>

Clustering-based ML Spam filtering. Clustering....

- kNN proposed in³ is a distribution free method, - i.e. the data is not required to be drawn from a given probability distribution.
 - this property is very vital as in the actual scenario, nearly all of the applied data disobey the standard hypothetical postulations made (such as Gaussian mixture, linearly separable, and others).
 - here, the classification model is not built from data.....

³

<https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbors-algorithm/>

Clustering-based ML Spam filtering. Clustering....

- kNN proposed in³ is a distribution free method, - i.e. the data is not required to be drawn from a given probability distribution.
 - this property is very vital as in the actual scenario, nearly all of the applied data disobey the standard hypothetical postulations made (such as Gaussian mixture, linearly separable, and others).
 - here, the classification model is not built from data.....
 - rather classification is carried out by matching the test instance with K training examples and

³

<https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbors-algorithm/>

Clustering-based ML Spam filtering. Clustering....

- kNN proposed in³ is a distribution free method, - i.e. the data is not required to be drawn from a given probability distribution.
 - this property is very vital as in the actual scenario, nearly all of the applied data disobey the standard hypothetical postulations made (such as Gaussian mixture, linearly separable, and others).
 - here, the classification model is not built from data.....
 - rather classification is carried out by matching the test instance with K training examples and
 - decision is made as to which group it belongs to depending on the resemblance to K closest neighbors.

³

<https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbors-algorithm/>

Clustering-based ML Spam filtering. Clustering....

- kNN proposed in³ is a distribution free method, - i.e. the data is not required to be drawn from a given probability distribution.
 - this property is very vital as in the actual scenario, nearly all of the applied data disobey the standard hypothetical postulations made (such as Gaussian mixture, linearly separable, and others).
 - here, the classification model is not built from data.....
 - rather classification is carried out by matching the test instance with K training examples and
 - decision is made as to which group it belongs to depending on the resemblance to K closest neighbors.
- is termed as a lazy learner since the training data points is not used by it to perform generalization.

³

<https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbors-algorithm/>

Clustering-based ML Spam filtering. Clustering....

- kNN proposed in³ is a distribution free method, - i.e. the data is not required to be drawn from a given probability distribution.
 - this property is very vital as in the actual scenario, nearly all of the applied data disobey the standard hypothetical postulations made (such as Gaussian mixture, linearly separable, and others).
 - here, the classification model is not built from data.....
 - rather classification is carried out by matching the test instance with K training examples and
 - decision is made as to which group it belongs to depending on the resemblance to K closest neighbors.
- is termed as a lazy learner since the training data points is not used by it to perform generalization.
- That is, there is no obvious training stage and if it exists it is extremely small.

³

<https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbors-algorithm/>

Clustering-based ML Spam filtering. Clustering....

- kNN proposed in³ is a distribution free method, - i.e. the data is not required to be drawn from a given probability distribution.
 - this property is very vital as in the actual scenario, nearly all of the applied data disobey the standard hypothetical postulations made (such as Gaussian mixture, linearly separable, and others).
 - here, the classification model is not built from data.....
 - rather classification is carried out by matching the test instance with K training examples and
 - decision is made as to which group it belongs to depending on the resemblance to K closest neighbors.
- is termed as a lazy learner since the training data points is not used by it to perform generalization.
- That is, there is no obvious training stage and if it exists it is extremely small.
- the implication is that the algorithm has a moderately speedy training phase.

³

<https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbors-algorithm/>

Clustering-based ML Spam filtering. Clustering....

- kNN proposed in³ is a distribution free method, - i.e. the data is not required to be drawn from a given probability distribution.
 - this property is very vital as in the actual scenario, nearly all of the applied data disobey the standard hypothetical postulations made (such as Gaussian mixture, linearly separable, and others).
 - here, the classification model is not built from data.....
 - rather classification is carried out by matching the test instance with K training examples and
 - decision is made as to which group it belongs to depending on the resemblance to K closest neighbors.
- is termed as a lazy learner since the training data points is not used by it to perform generalization.
- That is, there is no obvious training stage and if it exists it is extremely small.
- the implication is that the algorithm has a moderately speedy training phase.
- however, the entire training data is required throughout the testing phase as decisions are made based on the complete training data set.

³

<https://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/>

ML-based Spam filtering: Clustering: kNN algorithm pseudocode

Here Neighbours(d) return the k nearest neighbours of d , Closest (d, t) return the closest elements of t in d , and testClass(S) return the class label of S .

Algorithm 1 kNN Algorithm for Spam Email Classification

```
1: Find Email Message class labels.  
2: Input  $k$ , the number of nearest neighbors  
3: Input  $D$ , the set of test Email Message;  
4: Input  $T$ , the set of training Email Message.  
5:  $L$ , the label set of test Email Message.  
6: Read DataFile (TrainingData)  
7: Read DataFile (TestingData)  
8: for each  $d$  in  $D$  and each  $t$  in  $T$  do  
9:  $\text{Neighbors}(d) = \{\}$   
10: if  $|\text{Neighbors}(d)| < k$  then  
11:  $\text{Neighbors}(d) = \text{Closest}(d, t) \cup \text{Neighbors}(d)$   
12: end if  
13: if  $|\text{Neighbors}(d)| \geq k$  then  
14:  $\text{restrain}(M, x_j, y_j)$   
15: end if  
16: end for 17: return Final Email Message Classification (Spam/Valid email)  
18: end
```

1

Figure: KNN based Clustering for Spam detection

¹S. Zhu, W. Dong, W. Liu, Hierarchical reinforcement learning based on KNN classification algorithms, Int. J. Hosp. Inf. Technol. 8 (8) (2015) 175–184.

ML-based Spam filtering: Naïve Bayes classifier

Naïve Bayes classifier-based ML Spam filtering...The Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.

ML-based Spam filtering: Naïve Bayes classifier

Naïve Bayes classifier-based ML Spam filtering...The Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.
- acts as a fundamental probabilistic model

ML-based Spam filtering: Naïve Bayes classifier

Naïve Bayes classifier-based ML Spam filtering...The Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.
- acts as a fundamental probabilistic model
- assumes that the presence of one particular feature in a class doesn't affect the presence of another one i.e. all the predictors used as a feature are independent.

ML-based Spam filtering: Naïve Bayes classifier

Naïve Bayes classifier-based ML Spam filtering...The Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.
- acts as a fundamental probabilistic model
- assumes that the presence of one particular feature in a class doesn't affect the presence of another one i.e. all the predictors used as a feature are independent.
 - orange as fruit : round, orange and 3.5" of diameter - all contribute independently to a fruit being an orange.

ML-based Spam filtering: Naïve Bayes classifier

Naïve Bayes classifier-based ML Spam filtering...The Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.
- acts as a fundamental probabilistic model
- assumes that the presence of one particular feature in a class doesn't affect the presence of another one i.e. all the predictors used as a feature are independent.
 - orange as fruit : round, orange and 3.5" of diameter - all contribute independently to a fruit being an orange.
- hence, allows one to capture ambiguity about the model in an ethical way by influencing the probabilities of the results.

ML-based Spam filtering: Naïve Bayes classifier

Naïve Bayes classifier-based ML Spam filtering...The Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.
- acts as a fundamental probabilistic model
- assumes that the presence of one particular feature in a class doesn't affect the presence of another one i.e. all the predictors used as a feature are independent.
 - orange as fruit : round, orange and 3.5" of diameter - all contribute independently to a fruit being an orange.
- hence, allows one to capture ambiguity about the model in an ethical way by influencing the probabilities of the results.
- also called **the belief network**, uses **factored joint probability distribution** in a graphical model for decisions about **uncertain variables**

ML-based Spam filtering: Naïve Bayes classifier

Naïve Bayes classifier-based ML Spam filtering...The Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.
- acts as a fundamental probabilistic model
- assumes that the presence of one particular feature in a class doesn't affect the presence of another one i.e. all the predictors used as a feature are independent.
 - orange as fruit : round, orange and 3.5" of diameter - all contribute independently to a fruit being an orange.
- hence, allows one to capture ambiguity about the model in an ethical way by influencing the probabilities of the results.
- also called **the belief network**, uses **factored joint probability distribution** in a graphical model for decisions about **uncertain variables**
- Why do we have to resort to the probabilities for classification? Let us try to review/understand Bayes classification

Naïve Bayes classifier...: Conditional Probabilities

The usecase of fraudulent orders detection in an online store and the cost associated with the detection.

- It is emphasized that the cost of fraudulent transactions detection must be less than the cost of the fraud itself.

Naïve Bayes classifier...: Conditional Probabilities

The usecase of fraudulent orders detection in an online store and the cost associated with the detection.

- It is emphasized that the cost of fraudulent transactions detection must be less than the cost of the fraud itself.
- What if such is not the case ? e.g.

Naïve Bayes classifier...: Conditional Probabilities

The usecase of fraudulent orders detection in an online store and the cost associated with the detection.

- It is emphasized that the cost of fraudulent transactions detection must be less than the cost of the fraud itself.
- What if such is not the case ? e.g.
 - let us assume that about 10% of all orders coming in are fraudulent.

Naïve Bayes classifier...: Conditional Probabilities

The usecase of fraudulent orders detection in an online store and the cost associated with the detection.

- It is emphasized that the cost of fraudulent transactions detection must be less than the cost of the fraud itself.
- What if such is not the case ? e.g.
 - let us assume that about 10% of all orders coming in are fraudulent.
 - suppose one receives at least 1,000 orders per month,every single one were to be checked to ascertain that is a genuine transaction.

Naïve Bayes classifier...: Conditional Probabilities

The usecase of fraudulent orders detection in an online store and the cost associated with the detection.

- It is emphasized that the cost of fraudulent transactions detection must be less than the cost of the fraud itself.
- What if such is not the case ? e.g.
 - let us assume that about 10% of all orders coming in are fraudulent.
 - suppose one receives at least 1,000 orders per month,every single one were to be checked to ascertain that is a genuine transaction.
 - it takes up to 60 seconds per order to determine whether it's fraudulent or not,

Naïve Bayes classifier...: Conditional Probabilities

The usecase of fraudulent orders detection in an online store and the cost associated with the detection.

- It is emphasized that the cost of fraudulent transactions detection must be less than the cost of the fraud itself.
- What if such is not the case ? e.g.
 - let us assume that about 10% of all orders coming in are fraudulent.
 - suppose one receives at least 1,000 orders per month,every single one were to be checked to ascertain that is a genuine transaction.
 - it takes up to 60 seconds per order to determine whether it's fraudulent or not,
 - that it takes $12000 * 60 \text{ seconds/year} = 12000 \text{ minutes/year} = 200 \text{ hours/year}$.

Naïve Bayes classifier...: Conditional Probabilities

The usecase of fraudulent orders detection in an online store and the cost associated with the detection.

- It is emphasized that the cost of fraudulent transactions detection must be less than the cost of the fraud itself.
- What if such is not the case ? e.g.
 - let us assume that about 10% of all orders coming in are fraudulent.
 - suppose one receives at least 1,000 orders per month,every single one were to be checked to ascertain that is a genuine transaction.
 - it takes up to 60 seconds per order to determine whether it's fraudulent or not,
 - that it takes $12000 * 60 \text{ seconds/year} = 12000 \text{ minutes/year} = 200 \text{ hours/year}$.
 - a customer service representative costs around Rs 150 per hour to hire

Naïve Bayes classifier...: Conditional Probabilities

The usecase of fraudulent orders detection in an online store and the cost associated with the detection.

- It is emphasized that the cost of fraudulent transactions detection must be less than the cost of the fraud itself.
- What if such is not the case ? e.g.
 - let us assume that about 10% of all orders coming in are fraudulent.
 - suppose one receives at least 1,000 orders per month, every single one were to be checked to ascertain that is a genuine transaction.
 - it takes up to 60 seconds per order to determine whether it's fraudulent or not,
 - that it takes $12000 * 60 \text{ seconds/year} = 12000 \text{ minutes/year} = 200 \text{ hours/year}$.
 - a customer service representative costs around Rs 150 per hour to hire
 - then, the amount spent on fraud detection is Rs 30,000 per year.

Naïve Bayes classifier...: Conditional Probabilities

The usecase of fraudulent orders detection in an online store and the cost associated with the detection.

- It is emphasized that the cost of fraudulent transactions detection must be less than the cost of the fraud itself.
- What if such is not the case ? e.g.
 - let us assume that about 10% of all orders coming in are fraudulent.
 - suppose one receives at least 1,000 orders per month,every single one were to be checked to ascertain that is a genuine transaction.
 - it takes up to 60 seconds per order to determine whether it's fraudulent or not,
 - that it takes $12000 * 60 \text{ seconds/year} = 12000 \text{ minutes/year} = 200 \text{ hours/year}$.
 - a customer service representative costs around Rs 150 per hour to hire
 - then, the amount spent on fraud detection is Rs 30,000 per year.
- Can we find out what is the probability that an order is over 50% fraudulent? That would help reduce the number of orders to be examined.

Naïve Bayes classifier...: Conditional Probabilities

The usecase of fraudulent orders detection in an online store and the cost associated with the detection.

- It is emphasized that the cost of fraudulent transactions detection must be less than the cost of the fraud itself.
- What if such is not the case ? e.g.
 - let us assume that about 10% of all orders coming in are fraudulent.
 - suppose one receives at least 1,000 orders per month,every single one were to be checked to ascertain that is a genuine transaction.
 - it takes up to 60 seconds per order to determine whether it's fraudulent or not,
 - that it takes $12000 * 60 \text{ seconds/year} = 12000 \text{ minutes/year} = 200 \text{ hours/year}$.
 - a customer service representative costs around Rs 150 per hour to hire
 - then, the amount spent on fraud detection is Rs 30,000 per year.
- Can we find out what is the probability that an order is over 50% fraudulent? That would help reduce the number of orders to be examined.
- But here that is not the case, as we know here that only 10% of all orders coming in are fraudulent. So, how to proceed?

Naïve Bayes classifier...: Why probabilities?

- Can we compute a better alternative viz. what is the probability that **an order that often uses gift cards and multiple promotional codes is fraudulent ?**

¹<https://www.oreilly.com/library/view/thoughtful-machine-learning/9781449374075/ch04.html>

Naïve Bayes classifier...: Why probabilities?

- Can we compute a better alternative viz. what is the probability that **an order that often uses gift cards and multiple promotional codes is fraudulent ?**
- That is where **the conditional probabilities** and the consequence of the same viz. **Bayes Theorem & its adaptation by Pierre-Simon Laplace** come to our rescue.....

¹<https://www.oreilly.com/library/view/thoughtful-machine-learning/9781449374075/ch04.html>

Naïve Bayes classifier...: Why probabilities?

- Can we compute a better alternative viz. what is the probability that **an order that often uses gift cards and multiple promotional codes is fraudulent ?**
- That is where **the conditional probabilities** and the consequence of the same viz. **Bayes Theorem & its adaptation by Pierre-Simon Laplace** come to our rescue.....
- The conditional probabilities given as follows: The probability of A given that B happened in the past: i.e. Prob (A given B) is denoted as and is given by

$$\text{Prob}(A|B) = \frac{\text{Prob}(A \text{ and } B)}{\text{Prob}(B)}$$

¹<https://www.oreilly.com/library/view/thoughtful-machine-learning/9781449374075/ch04.html>

Naïve Bayes classifier...: Why probabilities?

- Can we compute a better alternative viz. what is the probability that **an order that often uses gift cards and multiple promotional codes is fraudulent ?**
- That is where **the conditional probabilities** and the consequence of the same viz. **Bayes Theorem & its adaptation by Pierre-Simon Laplace** come to our rescue.....
- The conditional probabilities given as follows: The probability of A given that B happened in the past: i.e. Prob (A given B) is denoted as and is given by

$$\text{Prob}(A|B) = \frac{\text{Prob}(A \text{ and } B)}{\text{Prob}(B)}$$

- This is illustrated here:

¹<https://www.oreilly.com/library/view/thoughtful-machine-learning/9781449374075/ch04.html>

Naïve Bayes classifier...: Why probabilities?

- Can we compute a better alternative viz. what is the probability that **an order that often uses gift cards and multiple promotional codes is fraudulent ?**
- That is where **the conditional probabilities** and the consequence of the same viz. **Bayes Theorem & its adaptation by Pierre-Simon Laplace** come to our rescue.....
- The conditional probabilities given as follows: The probability of A given that B happened in the past: i.e. Prob (A given B) is denoted as and is given by

$$\text{Prob}(A|B) = \frac{\text{Prob}(A \text{ and } B)}{\text{Prob}(B)}$$

- This is illustrated here:

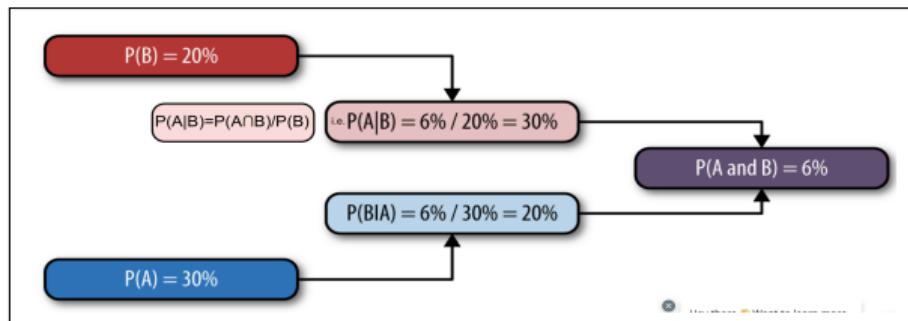
¹<https://www.oreilly.com/library/view/thoughtful-machine-learning/9781449374075/ch04.html>

Naïve Bayes classifier...: Why probabilities?

- Can we compute a better alternative viz. what is the probability that **an order that often uses gift cards and multiple promotional codes is fraudulent ?**
- That is where **the conditional probabilities** and the consequence of the same viz. **Bayes Theorem & its adaptation by Pierre-Simon Laplace** come to our rescue.....
- The conditional probabilities given as follows: The probability of A given that B happened in the past: i.e. Prob (A given B) is denoted as and is given by

$$Prob(A|B) = \frac{Prob(A \text{ and } B)}{Prob(B)}$$

- This is illustrated here:



1

Figure: How $P(A|B)$ sits between $P(A \text{ and } B)$ and $P(B)$

¹<https://www.oreilly.com/library/view/thoughtful-machine-learning/9781449374075/ch04.html>

Naïve Bayes classifier...: Why probabilities?...

- Thus, to measure the probability of fraud given that an order used a gift card would be given by

$$Prob(Fraud \mid Giftcard) = \frac{Prob(Fraud \cap Giftcard)}{Prob(Giftcard)}$$

Naïve Bayes classifier...: Why probabilities?...

- Thus, to measure the probability of fraud given that an order used a gift card would be given by

$$\text{Prob}(\text{Fraud} \mid \text{Giftcard}) = \frac{\text{Prob}(\text{Fraud} \cap \text{Giftcard})}{\text{Prob}(\text{Giftcard})}$$

- Now this can be solved if one can compute the actual probability of Fraud and Giftcard happening.

Naïve Bayes classifier...: Why probabilities?...

- Thus, to measure the probability of fraud given that an order used a gift card would be given by

$$Prob(Fraud \mid Giftcard) = \frac{Prob(Fraud \cap Giftcard)}{Prob(Giftcard)}$$

- Now this can be solved if one can compute the actual probability of Fraud and Giftcard happening.
- But, computing $P(Fraud \cap Giftcard)$ is not obvious.

Naïve Bayes classifier...: Why probabilities?...

- Thus, to measure the probability of fraud given that an order used a gift card would be given by

$$Prob(Fraud \mid Giftcard) = \frac{Prob(Fraud \cap Giftcard)}{Prob(Giftcard)}$$

- Now this can be solved if one can compute the actual probability of Fraud and Giftcard happening.
- But, computing $P(Fraud \cap Giftcard)$ is not obvious.
- This is where the **Bayes Theorem & its adaptation by Pierre-Simon Laplace** come to our rescue..... viz. expression for Inverse Conditional Probability (aka Bayes' Theorem) as follows:

Naïve Bayes classifier...: Why probabilities?...

- Thus, to measure the probability of fraud given that an order used a gift card would be given by

$$Prob(Fraud \mid Giftcard) = \frac{Prob(Fraud \cap Giftcard)}{Prob(Giftcard)}$$

- Now this can be solved if one can compute the actual probability of Fraud and Giftcard happening.
- But, computing $P(Fraud \cap Giftcard)$ is not obvious.
- This is where the **Bayes Theorem & its adaptation by Pierre-Simon Laplace** come to our rescue..... viz. expression for Inverse Conditional Probability (aka Bayes' Theorem) as follows:

Naïve Bayes classifier...: Why probabilities?...

- Thus, to measure the probability of fraud given that an order used a gift card would be given by

$$\text{Prob}(\text{Fraud} \mid \text{Giftcard}) = \frac{\text{Prob}(\text{Fraud} \cap \text{Giftcard})}{\text{Prob}(\text{Giftcard})}$$

- Now this can be solved if one can compute the actual probability of Fraud and Giftcard happening.
- But, computing $\text{P}(\text{Fraud} \cap \text{Giftcard})$ is not obvious.
- This is where the **Bayes Theorem & its adaptation by Pierre-Simon Laplace** come to our rescue..... viz. expression for Inverse Conditional Probability (aka Bayes' Theorem) as follows:

Thus, A Naïve Bayes classifier

is a straightforward probabilistic classifier that is founded on Bayes theorem with sound assumptions that are independent in nature. The expression for the probability model - that should be autonomous characteristic model is given as

$$\text{Bayes Theorem : } \text{Prob}(B \mid A) = \frac{\text{Prob}(A \mid B)\text{Prob}(B)}{\text{Prob}(A)}$$

This result can be understood as follows:

$$\begin{aligned} \text{Bayes Theorem : } \text{Prob}(B | A) &= \frac{\text{Prob}(A | B)\text{Prob}(B)}{\text{P}(A)} \\ &= \frac{\frac{\text{Prob}(A \cap B)\text{P}(B)}{\text{P}(B)}}{\text{P}(A)} = \frac{\text{Prob}(A \cap B)}{\text{P}(A)} \end{aligned}$$

This result can be understood as follows:

$$\begin{aligned} \text{Bayes Theorem : } \text{Prob}(B | A) &= \frac{\text{Prob}(A | B)\text{Prob}(B)}{\text{P}(A)} \\ &= \frac{\frac{\text{Prob}(A \cap B)\text{P}(B)}{\text{P}(B)}}{\text{P}(A)} = \frac{\text{Prob}(A \cap B)}{\text{P}(A)} \end{aligned}$$

This is useful in our fraud detection example as follows:

$$\begin{aligned} P(\text{Fraud} | \text{Giftcard}) &= \frac{\text{Prob}(\text{Giftcard} | \text{Fraud})\text{P}(\text{Fraud})}{\text{P}(\text{FGiftcard})} \\ P(\text{Fraud} | \text{Giftcard}) &= \frac{\text{Prob}(\text{Giftcard} \cap \text{Fraud})}{\text{P}(\text{FGiftcard})} \end{aligned}$$

This result can be understood as follows:

$$\text{Bayes Theorem : } \text{Prob}(B | A) = \frac{\text{Prob}(A | B)\text{Prob}(B)}{\text{Prob}(A)}$$

$$= \frac{\frac{\text{Prob}(A \cap B)\text{Prob}(B)}{\text{Prob}(B)}}{\text{Prob}(A)} = \frac{\text{Prob}(A \cap B)}{\text{Prob}(A)}$$

This is useful in our fraud detection example as follows:

$$P(\text{Fraud} | \text{Giftcard}) = \frac{\text{Prob}(\text{Giftcard} | \text{Fraud})\text{Prob}(\text{Fraud})}{\text{Prob}(\text{FGiftcard})}$$

$$P(\text{Fraud} | \text{Giftcard}) = \frac{\text{Prob}(\text{Giftcard} \cap \text{Fraud})}{\text{Prob}(\text{FGiftcard})}$$

Thus, if $P(\text{Fraud})=10\%$. $P(\text{Giftcard}) = 10\%$, and based on our research, we know that $P(\text{Giftcard} | \text{Fraud})$ i.e. the probability of gift card use in a fraudulent order = 60%, then,

$$P(\text{Fraud} | \text{Giftcard}) = \frac{60 * 10}{10} = 60\%$$

This result can be understood as follows:

$$\text{Bayes Theorem : } \text{Prob}(B | A) = \frac{\text{Prob}(A | B)\text{Prob}(B)}{\text{P}(A)}$$

$$= \frac{\frac{\text{Prob}(A \cap B)\text{P}(B)}{\text{P}(B)}}{\text{P}(A)} = \frac{\text{Prob}(A \cap B)}{\text{P}(A)}$$

This is useful in our fraud detection example as follows:

$$P(\text{Fraud} | \text{Giftcard}) = \frac{\text{Prob}(\text{Giftcard} | \text{Fraud})\text{P}(\text{Fraud})}{\text{P}(\text{FGiftcard})}$$

$$P(\text{Fraud} | \text{Giftcard}) = \frac{\text{Prob}(\text{Giftcard} \cap \text{Fraud})}{\text{P}(\text{FGiftcard})}$$

Thus, if $\text{P}(\text{Fraud})=10\%$. $\text{P}(\text{Giftcard}) = 10\%$, and based on our research, we know that $\text{P}(\text{Giftcard} | \text{Fraud})$ i.e. the probability of gift card use in a fraudulent order = 60%, then,

$$P(\text{Fraud} | \text{Giftcard}) = \frac{60 * 10}{10} = 60\%$$

Naïve Bayes classifier...: Bayesian Network

Typical BNs are presented with

- nodes representing random variables

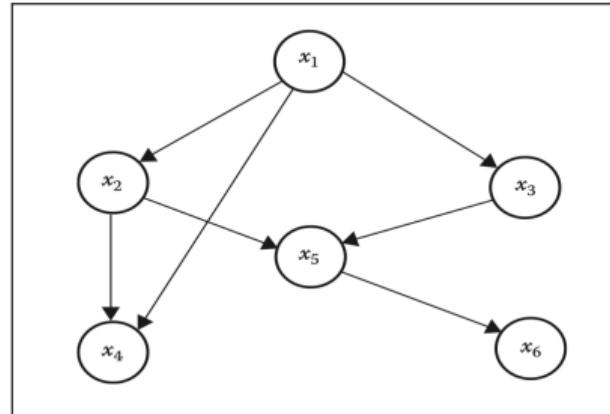


Figure: BN with sample factored joint distribution.

Here,

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5, x_6) \\ = P(x_6|x_5)P(x_5|x_3, x_2) \\ P(x_4|x_2, x_1)P(x_3|x_1) \\ P(x_2|x_1)P(x_1) \end{aligned}$$

Naïve Bayes classifier...: Bayesian Network

Typical BNs are presented with

- nodes representing random variables
- arcs representing

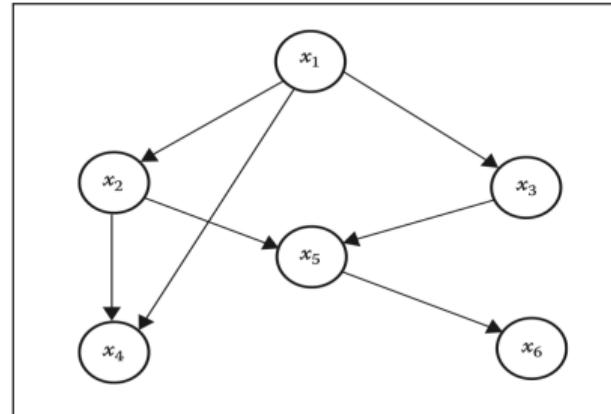


Figure: BN with sample factored joint distribution.

Here,

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5, x_6) \\ = P(x_6|x_5)P(x_5|x_3, x_2) \\ P(x_4|x_2, x_1)P(x_3|x_1) \\ P(x_2|x_1)P(x_1) \end{aligned}$$

Naïve Bayes classifier...: Bayesian Network

Typical BNs are presented with

- nodes representing random variables
- arcs representing
 - probabilistic dependencies between variables, and

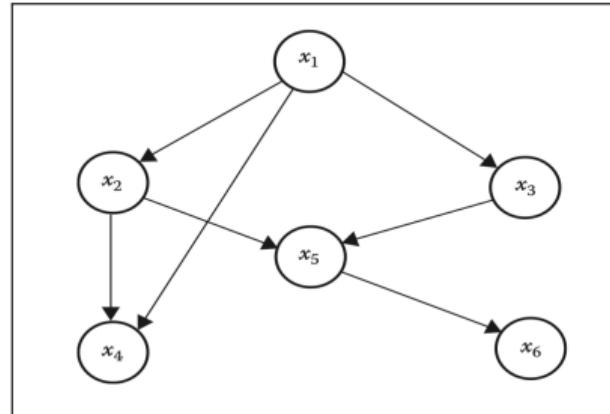


Figure: BN with sample factored joint distribution.

Here,

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5, x_6) \\ = P(x_6|x_5)P(x_5|x_3, x_2) \\ P(x_4|x_2, x_1)P(x_3|x_1) \\ P(x_2|x_1)P(x_1) \end{aligned}$$

Naïve Bayes classifier...: Bayesian Network

Typical BNs are presented with

- nodes representing random variables
- arcs representing
 - probabilistic dependencies between variables, and
 - conditional probabilities encoding the strength of the dependencies,

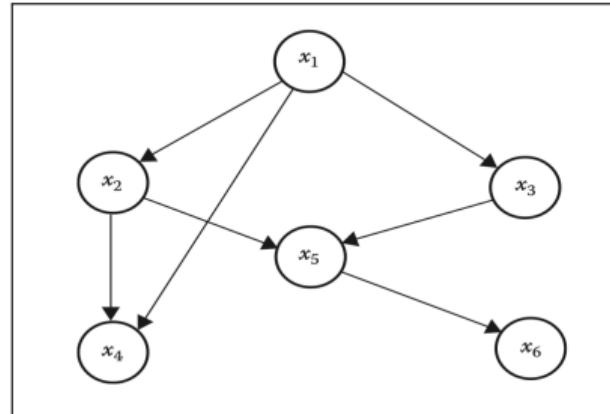


Figure: BN with sample factored joint distribution.

Here,

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5, x_6) \\ = P(x_6|x_5)P(x_5|x_3, x_2) \\ P(x_4|x_2, x_1)P(x_3|x_1) \\ P(x_2|x_1)P(x_1) \end{aligned}$$

Naïve Bayes classifier...: Bayesian Network

Typical BNs are presented with

- nodes representing random variables
- arcs representing
 - probabilistic dependencies between variables, and
 - conditional probabilities encoding the strength of the dependencies,
- while unconnected nodes refer to variables that are independent of each other.

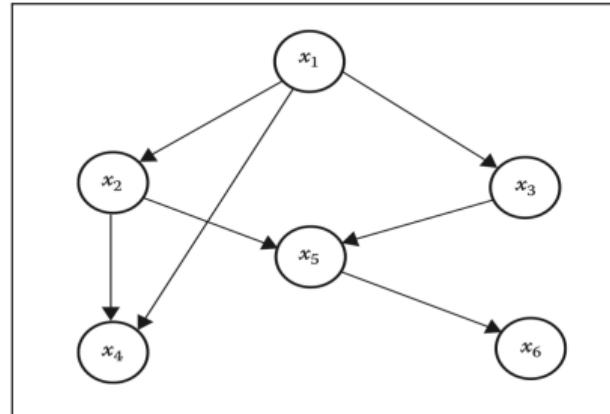


Figure: BN with sample factored joint distribution.

Here,

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5, x_6) \\ = P(x_6|x_5)P(x_5|x_3, x_2) \\ P(x_4|x_2, x_1)P(x_3|x_1) \\ P(x_2|x_1)P(x_1) \end{aligned}$$

Naïve Bayes classifier...: Bayesian Network

Typical BNs are presented with

- nodes representing random variables
- arcs representing
 - probabilistic dependencies between variables, and
 - conditional probabilities encoding the strength of the dependencies,
- while unconnected nodes refer to variables that are independent of each other.
- each node is associated with a probability function corresponding to the node's parent variables.

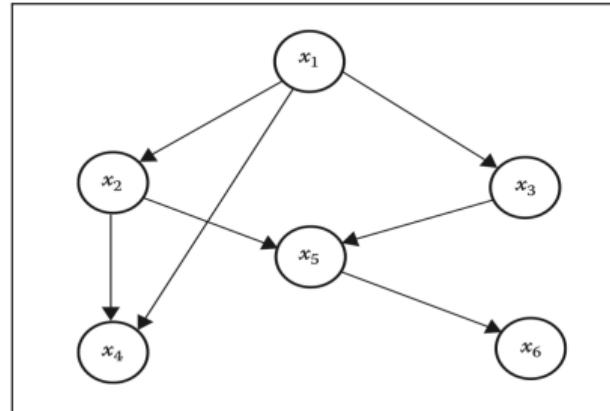


Figure: BN with sample factored joint distribution.

Here,

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5, x_6) \\ = P(x_6|x_5)P(x_5|x_3, x_2) \\ P(x_4|x_2, x_1)P(x_3|x_1) \\ P(x_2|x_1)P(x_1) \end{aligned}$$

Naïve Bayes classifier...: Bayesian Network

Typical BNs are presented with

- nodes representing random variables
- arcs representing
 - probabilistic dependencies between variables, and
 - conditional probabilities encoding the strength of the dependencies,
- while unconnected nodes refer to variables that are independent of each other.
- each node is associated with a probability function corresponding to the node's parent variables.
- the node always computes posterior probabilities given proof of the parents for the selected nodes.

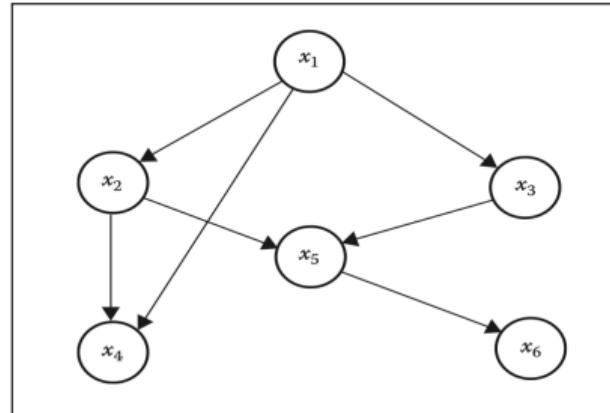


Figure: BN with sample factored joint distribution.

Here,

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5, x_6) \\ = P(x_6|x_5)P(x_5|x_3, x_2) \\ P(x_4|x_2, x_1)P(x_3|x_1) \\ P(x_2|x_1)P(x_1) \end{aligned}$$

Naïve Bayes classifier...: Bayesian Network

Typical BNs are presented with

- nodes representing random variables
- arcs representing
 - probabilistic dependencies between variables, and
 - conditional probabilities encoding the strength of the dependencies,
- while unconnected nodes refer to variables that are independent of each other.
- each node is associated with a probability function corresponding to the node's parent variables.
- the node always computes posterior probabilities given proof of the parents for the selected nodes.
- e.g. in the figure here....

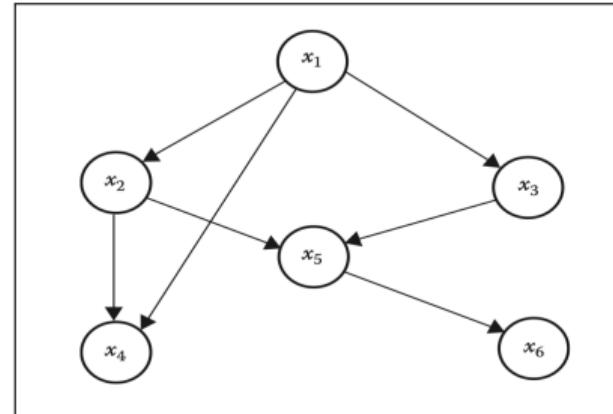


Figure: BN with sample factored joint distribution.

Here,

$$\begin{aligned} P(x_1, x_2, x_3, x_4, x_5, x_6) \\ = P(x_6|x_5)P(x_5|x_3, x_2) \\ P(x_4|x_2, x_1)P(x_3|x_1) \\ P(x_2|x_1)P(x_1) \end{aligned}$$

Naïve Bayes classifier

Thus, the Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.

Naïve Bayes classifier

Thus, the Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.
- acts as a fundamental probabilistic model

Thus, the Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.
- acts as a fundamental probabilistic model
- hence, allows one to capture ambiguity about the model in an ethical way by influencing the probabilities of the results.

Thus, the Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.
- acts as a fundamental probabilistic model
- hence, allows one to capture ambiguity about the model in an ethical way by influencing the probabilities of the results.
- is used to provide solution to analytical and predictive problems.

Thus, the Bayesian classification exemplifies ...

- a supervised learning technique and at the same time a statistical technique for classification.
- acts as a fundamental probabilistic model
- hence, allows one to capture ambiguity about the model in an ethical way by influencing the probabilities of the results.
- is used to provide solution to analytical and predictive problems.
- computes exact likelihoods for postulation and it is robust to noise in input data.

The Bayesian classification ...

- The notion of class restrictive autonomy in Bayes classifier ? What is it and Why was it created ?

⁴ M.N. Marsono et al, Binary LNS-Based Naïve Bayes Inference Engine for Spam Control: Noise Analysis and FPGA Synthesis, IET Computers & Digital Techniques, 2008.

The Bayesian classification ...

- The notion of class restrictive autonomy in Bayes classifier ? What is it and Why was it created ?
- Why is the classifier called as 'naïve' ?

⁴ M.N. Marsono et al, Binary LNS-Based Naïve Bayes Inference Engine for Spam Control: Noise Analysis and FPGA Synthesis, IET Computers & Digital Techniques, 2008.

Naïve Bayes classifier...

The Bayesian classification ...

- The notion of class restrictive autonomy in Bayes classifier ? What is it and Why was it created ?
- Why is the classifier called as ‘naïve’ ?
- greatly accepted as a simple and computationally efficient algorithm with satisfactory performances in solving real-world problems.

⁴ M.N. Marsono et al, Binary LNS-Based Naïve Bayes Inference Engine for Spam Control: Noise Analysis and FPGA Synthesis, IET Computers & Digital Techniques, 2008.

The Bayesian classification ...

- The notion of class restrictive autonomy in Bayes classifier ? What is it and Why was it created ?
- Why is the classifier called as ‘naïve’ ?
- greatly accepted as a simple and computationally efficient algorithm with satisfactory performances in solving real-world problems.
- due to this obvious advantage, it is extensively applied in the field of spam filtering (detect spam e-mail) and sentiment analysis (in social media analysis, to recognize positive and negative customer opinions).

⁴ M.N. Marsono et al, Binary LNS-Based Naïve Bayes Inference Engine for Spam Control: Noise Analysis and FPGA Synthesis, IET Computers & Digital Techniques, 2008.

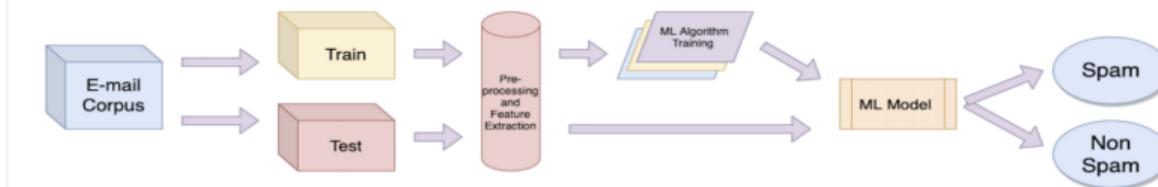
The Bayesian classification ...

- The notion of class restrictive autonomy in Bayes classifier ? What is it and Why was it created ?
- Why is the classifier called as 'naïve' ?
- greatly accepted as a simple and computationally efficient algorithm with satisfactory performances in solving real-world problems.
- due to this obvious advantage, it is extensively applied in the field of spam filtering (detect spam e-mail) and sentiment analysis (in social media analysis, to recognize positive and negative customer opinions).
- some authors ⁴ claim that virtually all the statistic-based Spam filtering techniques use Naïve Bayes' classifier

⁴ M.N. Marsono et al, Binary LNS-Based Naïve Bayes Inference Engine for Spam Control: Noise Analysis and FPGA Synthesis, IET Computers & Digital Techniques, 2008.

ML-based Spam filtering: Using Naïve Bayes classifier

Training Testing Phase



New Email Classification

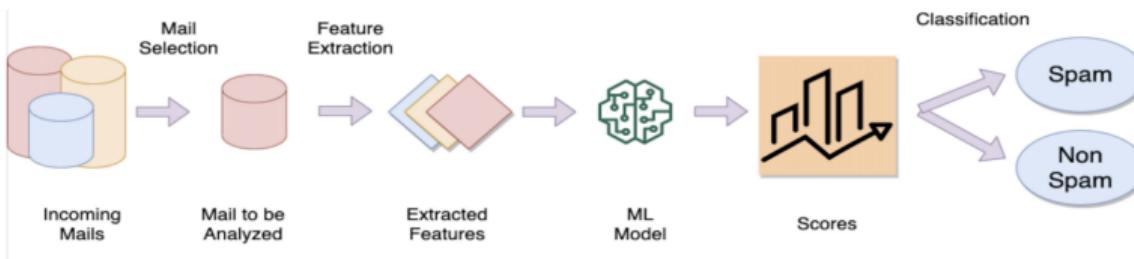


Figure: Typical ML based Spam filtering schematic

1

Fig Src.: <https://www.enjoyalgorithms.com/blog/email-spam-and-non-spam-filtering-using-machine-learning>

ML-based Spam filtering: Using Naïve Bayes classifier

First the formal definition of Naïve Bayes classifier based Spam filtering

- Given a set of messages $\mathbb{M} = m_1, m_2, \dots, m_j, \dots, m_{\mathbb{M}}$ and category set $\mathbb{C} = \text{spam}(c_s), \text{legitimate}(c_l)$, where m_j is the j^{th} mail in \mathbb{M} and \mathbb{C} is the possible label set, the task of automated Spam filtering consists in building a Boolean categorization function $\phi(m_j, c_i) : M \times C \rightarrow \{\text{True}, \text{False}\}$. When $\phi(m_j, c_i)$ is True, it indicates that message m_j belongs to category c_i ; otherwise, m_j does not belong to c_i .

ML-based Spam filtering: Using Naïve Bayes classifier

First the formal definition of Naïve Bayes classifier based Spam filtering

- Given a set of messages $\mathbb{M} = m_1, m_2, \dots, m_j, \dots, m_{\mathbb{M}}$ and category set $\mathbb{C} = \text{spam}(c_s), \text{legitimate}(c_l)$, where m_j is the j^{th} mail in \mathbb{M} and \mathbb{C} is the possible label set, the task of automated Spam filtering consists in building a Boolean categorization function $\phi(m_j, c_i) : M \times C \rightarrow \{\text{True}, \text{False}\}$. When $\phi(m_j, c_i)$ is True, it indicates that message m_j belongs to category c_i ; otherwise, m_j does not belong to c_i .
- The steps to be followed in the implementation and execution of such a classifier are as follows:

ML-based Spam filtering: Using Naïve Bayes classifier

First the formal definition of Naïve Bayes classifier based Spam filtering

- Given a set of messages $\mathbb{M} = m_1, m_2, \dots, m_j, \dots, m_{\mathbb{M}}$ and category set $\mathbb{C} = \text{spam}(c_s), \text{legitimate}(c_l)$, where m_j is the j^{th} mail in \mathbb{M} and \mathbb{C} is the possible label set, the task of automated Spam filtering consists in building a Boolean categorization function $\phi(m_j, c_i) : M \times C \rightarrow \{\text{True}, \text{False}\}$. When $\phi(m_j, c_i)$ is True, it indicates that message m_j belongs to category c_i ; otherwise, m_j does not belong to c_i .
- The steps to be followed in the implementation and execution of such a classifier are as follows:
 - Email Data collection

ML-based Spam filtering: Using Naïve Bayes classifier

First the formal definition of Naïve Bayes classifier based Spam filtering

- Given a set of messages $\mathbb{M} = m_1, m_2, \dots, m_j, \dots, m_{\mathbb{M}}$ and category set $\mathbb{C} = \text{spam}(c_s), \text{legitimate}(c_l)$, where m_j is the j^{th} mail in \mathbb{M} and \mathbb{C} is the possible label set, the task of automated Spam filtering consists in building a Boolean categorization function $\phi(m_j, c_i) : M \times C \rightarrow \{\text{True}, \text{False}\}$. When $\phi(m_j, c_i)$ is True, it indicates that message m_j belongs to category c_i ; otherwise, m_j does not belong to c_i .
- The steps to be followed in the implementation and execution of such a classifier are as follows:
 - Email Data collection
 - Pre-processing of the Email content including extracting words from images

ML-based Spam filtering: Using Naïve Bayes classifier

First the formal definition of Naïve Bayes classifier based Spam filtering

- Given a set of messages $\mathbb{M} = m_1, m_2, \dots, m_j, \dots, m_{\mathbb{M}}$ and category set $\mathbb{C} = \text{spam}(c_s), \text{legitimate}(c_l)$, where m_j is the j^{th} mail in \mathbb{M} and \mathbb{C} is the possible label set, the task of automated Spam filtering consists in building a Boolean categorization function $\phi(m_j, c_i) : M \times C \rightarrow \{\text{True}, \text{False}\}$. When $\phi(m_j, c_i)$ is True, it indicates that message m_j belongs to category c_i ; otherwise, m_j does not belong to c_i .
- The steps to be followed in the implementation and execution of such a classifier are as follows:
 - Email Data collection
 - Pre-processing of the Email content including extracting words from images
 - Feature extraction and selection

ML-based Spam filtering: Using Naïve Bayes classifier

First the formal definition of Naïve Bayes classifier based Spam filtering

- Given a set of messages $\mathbb{M} = m_1, m_2, \dots, m_j, \dots, m_{\mathbb{M}}$ and category set $\mathbb{C} = \text{spam}(c_s), \text{legitimate}(c_l)$, where m_j is the j^{th} mail in \mathbb{M} and \mathbb{C} is the possible label set, the task of automated Spam filtering consists in building a Boolean categorization function $\phi(m_j, c_i) : M \times C \rightarrow \{\text{True}, \text{False}\}$. When $\phi(m_j, c_i)$ is True, it indicates that message m_j belongs to category c_i ; otherwise, m_j does not belong to c_i .
- The steps to be followed in the implementation and execution of such a classifier are as follows:
 - Email Data collection
 - Pre-processing of the Email content including extracting words from images
 - Feature extraction and selection
 - Naïve Bayes classifier implementation

ML-based Spam filtering: Using Naïve Bayes classifier

First the formal definition of Naïve Bayes classifier based Spam filtering

- Given a set of messages $\mathbb{M} = m_1, m_2, \dots, m_j, \dots, m_{\mathbb{M}}$ and category set $\mathbb{C} = \text{spam}(c_s), \text{legitimate}(c_l)$, where m_j is the j^{th} mail in \mathbb{M} and \mathbb{C} is the possible label set, the task of automated Spam filtering consists in building a Boolean categorization function $\phi(m_j, c_i) : M \times C \rightarrow \{\text{True}, \text{False}\}$. When $\phi(m_j, c_i)$ is True, it indicates that message m_j belongs to category c_i ; otherwise, m_j does not belong to c_i .
- The steps to be followed in the implementation and execution of such a classifier are as follows:
 - Email Data collection
 - Pre-processing of the Email content including extracting words from images
 - Feature extraction and selection
 - Naïve Bayes classifier implementation
 - Analysis to improve the performance

ML-based Spam filtering: Using Naïve Bayes classifier

First the formal definition of Naïve Bayes classifier based Spam filtering

- Given a set of messages $\mathbb{M} = m_1, m_2, \dots, m_j, \dots, m_{\mathbb{M}}$ and category set $\mathbb{C} = \text{spam}(c_s), \text{legitimate}(c_l)$, where m_j is the j^{th} mail in \mathbb{M} and \mathbb{C} is the possible label set, the task of automated Spam filtering consists in building a Boolean categorization function $\phi(m_j, c_i) : M \times C \rightarrow \{\text{True}, \text{False}\}$. When $\phi(m_j, c_i)$ is True, it indicates that message m_j belongs to category c_i ; otherwise, m_j does not belong to c_i .
- The steps to be followed in the implementation and execution of such a classifier are as follows:
 - Email Data collection
 - Pre-processing of the Email content including extracting words from images
 - Feature extraction and selection
 - Naïve Bayes classifier implementation
 - Analysis to improve the performance
- There is an open source tool SpamAssassin mail filter to identify Spam mails. It is an intelligent email filter which uses a diverse range of tests to identify undesirable email messages. It uses Bayesian Spam filter.

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Email Data collection:

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Email Data collection:
 - numerous sources of data that one can use, specifically those raw email messages marked as either spam or ham.

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Email Data collection:

- numerous sources of data that one can use, specifically those raw email messages marked as either spam or ham.
- e.g. CSDMC2010 SPAM corpus (SourceForge), Enron corpus (with 55% Spam), Trec (with 67% Spam)

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Email Data collection:
 - numerous sources of data that one can use, specifically those raw email messages marked as either spam or ham.
 - e.g. CSDMC2010 SPAM corpus (SourceForge), Enron corpus (with 55% Spam), Trec (with 67% Spam)
- Pre-processing the Email content:

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Email Data collection:
 - numerous sources of data that one can use, specifically those raw email messages marked as either spam or ham.
 - e.g. CSDMC2010 SPAM corpus (SourceForge), Enron corpus (with 55% Spam), Trec (with 67% Spam)
- Pre-processing the Email content:
 - One of the steps in pre-processing performed early is to parse an incoming email message into its subject i.e. the header and the body.

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Email Data collection:

- numerous sources of data that one can use, specifically those raw email messages marked as either spam or ham.
- e.g. CSDMC2010 SPAM corpus (SourceForge), Enron corpus (with 55% Spam), Trec (with 67% Spam)

- Pre-processing the Email content:

- One of the steps in pre-processing performed early is to parse an incoming email message into its subject i.e. the header and the body.
- At times, a software application is used for data extraction - extracts text data from the email header and body. It can also parse data directly from email file attachments like PDF documents, CSV files, and MS Office files.

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Email Data collection:

- numerous sources of data that one can use, specifically those raw email messages marked as either spam or ham.
- e.g. CSDMC2010 SPAM corpus (SourceForge), Enron corpus (with 55% Spam), Trec (with 67% Spam)

- Pre-processing the Email content:

- One of the steps in pre-processing performed early is to parse an incoming email message into its subject i.e. the header and the body.
- At times, a software application is used for data extraction - extracts text data from the email header and body. It can also parse data directly from email file attachments like PDF documents, CSV files, and MS Office files.
- the next step performed is almost always tokenization of email - a process where the content of an email is broken into words, a sequence of representative symbols called tokens.

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Email Data collection:
 - numerous sources of data that one can use, specifically those raw email messages marked as either spam or ham.
 - e.g. CSDMC2010 SPAM corpus (SourceForge), Enron corpus (with 55% Spam), Trec (with 67% Spam)
- Pre-processing the Email content:
 - One of the steps in pre-processing performed early is to parse an incoming email message into its subject i.e. the header and the body.
 - At times, a software application is used for data extraction - extracts text data from the email header and body. It can also parse data directly from email file attachments like PDF documents, CSV files, and MS Office files.
 - the next step performed is almost always tokenization of email - a process where the content of an email is broken into words, a sequence of representative symbols called tokens.
 - the tokens may be extracted from the email body, header, subject and images.

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Email Data collection:
 - numerous sources of data that one can use, specifically those raw email messages marked as either spam or ham.
 - e.g. CSDMC2010 SPAM corpus (SourceForge), Enron corpus (with 55% Spam), Trec (with 67% Spam)
- Pre-processing the Email content:
 - One of the steps in pre-processing performed early is to parse an incoming email message into its subject i.e. the header and the body.
 - At times, a software application is used for data extraction - extracts text data from the email header and body. It can also parse data directly from email file attachments like PDF documents, CSV files, and MS Office files.
 - the next step performed is almost always tokenization of email - a process where the content of an email is broken into words, a sequence of representative symbols called tokens.
 - the tokens may be extracted from the email body, header, subject and images.
 - the Google library Tesseract can allow extraction of the words from the images also automatically.

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Pre-processing the Email content...*continued*:

1

¹ Fig Src.:<https://www.oreilly.com/library/view/thoughtful-machine-learning/9781449374075/ch04.html>

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Pre-processing the Email content...*continued*:

- there are numerous ways to tokenize text, such as by stems, word frequencies, and words as shown below.

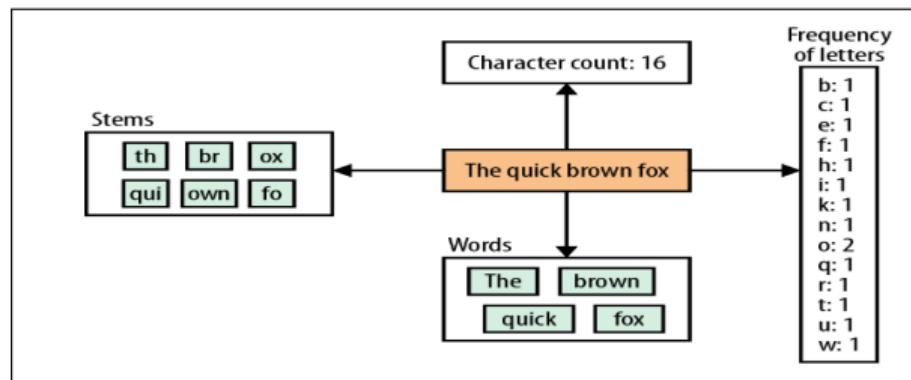


Figure: Different ways to tokenize

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Pre-processing the Email content...*continued*:

- there are numerous ways to tokenize text, such as by stems, word frequencies, and words as shown below.

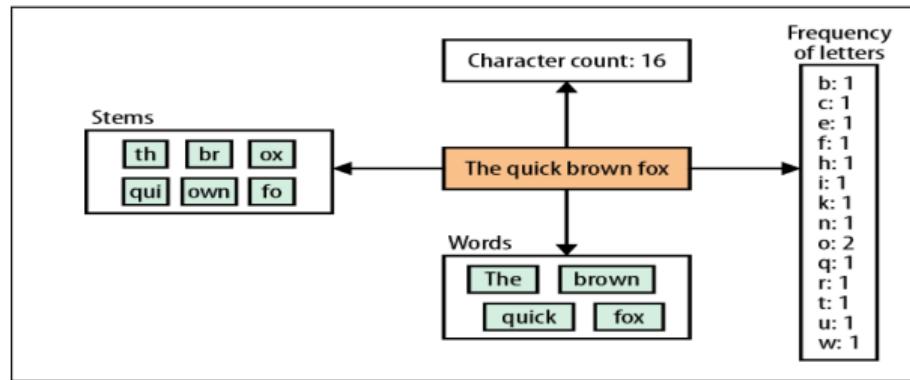


Figure: Different ways to tokenize

- tokenization also has to take care of phrases that are more contextual e.g. the phrase **Buy now** sounds spammy, whereas **Buy** and **now** do not.

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Pre-processing the Email content...*continued*:

- there are numerous ways to tokenize text, such as by stems, word frequencies, and words as shown below.

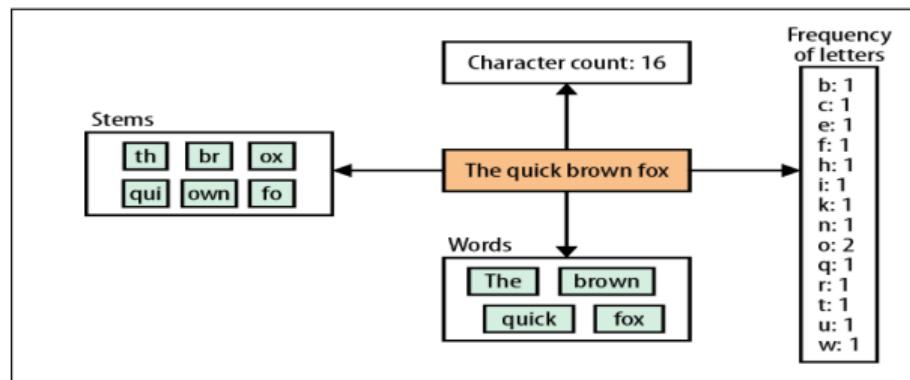


Figure: Different ways to tokenize

- tokenization also has to take care of phrases that are more contextual e.g. the phrase **Buy now** sounds spammy, whereas **Buy** and **now** do not.
- in Bayesian classification it is assumed that each individual token contributes to the **spamminess** of the email.

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Feature Extraction and Selection:

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Feature Extraction and Selection:

- the words that result after pre-processing can be stored in a database that maintains the frequencies of the different words represented in each column (shown in the figure before)

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Feature Extraction and Selection:

- the words that result after pre-processing can be stored in a database that maintains the frequencies of the different words represented in each column (shown in the figure before)
- Next the features or the attributes are to be extracted. This is done on the basis of either the

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Feature Extraction and Selection:

- the words that result after pre-processing can be stored in a database that maintains the frequencies of the different words represented in each column (shown in the figure before)
- Next the features or the attributes are to be extracted. This is done on the basis of either the
 - **Essential** attributes (adult content bag pf words, number of semantic discrepancies etc.),

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Feature Extraction and Selection:

- the words that result after pre-processing can be stored in a database that maintains the frequencies of the different words represented in each column (shown in the figure before)
- Next the features or the attributes are to be extracted. This is done on the basis of either the
 - **Essential attributes** (adult content bag pf words, number of semantic discrepancies etc.),
 - **Additional attributes** (sender country, IP, email, age number of replies, number of recipients etc.) and

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Feature Extraction and Selection:

- the words that result after pre-processing can be stored in a database that maintains the frequencies of the different words represented in each column (shown in the figure before)
- Next the features or the attributes are to be extracted. This is done on the basis of either the
 - **Essential attributes** (adult content bag pf words, number of semantic discrepancies etc.),
 - **Additional attributes** (sender country, IP, email, age number of replies, number of recepients etc.) and
 - **Less important attributes** (e.g. sender's DoB, Account lifespan, Age of the recipient etc.)

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Feature Extraction and Selection:

- the words that result after pre-processing can be stored in a database that maintains the frequencies of the different words represented in each column (shown in the figure before)
- Next the features or the attributes are to be extracted. This is done on the basis of either the
 - **Essential attributes** (adult content bag pf words, number of semantic discrepancies etc.),
 - **Additional attributes** (sender country, IP, email, age number of replies, number of recepients etc.) and
 - **Less important attributes** (e.g. sender's DoB, Account lifespan, Age of the recipient etc.)
- the techniques of stemming, noise removal, stop-word removal etc are useful in reducing the number of attributes.

ML-based Spam filtering: Using Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Feature Extraction and Selection:

- the words that result after pre-processing can be stored in a database that maintains the frequencies of the different words represented in each column (shown in the figure before)
 - Next the features or the attributes are to be extracted. This is done on the basis of either the
 - **Essential attributes** (adult content bag pf words, number of semantic discrepancies etc.),
 - **Additional attributes** (sender country, IP, email, age number of replies, number of recepients etc.) and
 - **Less important attributes** (e.g. sender's DoB, Account lifespan, Age of the recipient etc.)
 - the techniques of stemming, noise removal, stop-word removal etc are useful in reducing the number of attributes.
- Next step now is the implementation of the Bayesian classifier.....

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- these are based on grouping the statistics of each token to a total score.

⁵K. Li, Z. Zhong, Fast statistical spam filter by approximate classifications, In: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, 2006. Saint Malo, France.

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- these are based on grouping the statistics of each token to a total score.
- the score is used in making resolution on the filtering.

⁵K. Li, Z. Zhong, Fast statistical spam filter by approximate classifications, In: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, 2006. Saint Malo, France.

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- these are based on grouping the statistics of each token to a total score.
- the score is used in making resolution on the filtering.
- According to ⁵, the token T which denote the **spamminess (spam rating)** is computed as

$$S[T] = \frac{C_{Spam}(T)}{C_{Spam}(T) + C_{Ham}(T)}$$

⁵K. Li, Z. Zhong, Fast statistical spam filter by approximate classifications, In: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, 2006. Saint Malo, France.

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- these are based on grouping the statistics of each token to a total score.
- the score is used in making resolution on the filtering.
- According to⁵, the token T which denote the **spamminess (spam rating)** is computed as

$$S[T] = \frac{C_{Spam}(T)}{C_{Spam}(T) + C_{Ham}(T)}$$

- Here, $C_{Spam}(T)$ = The number of spam messages containing token T , and

⁵K. Li, Z. Zhong, Fast statistical spam filter by approximate classifications, In: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, 2006. Saint Malo, France.

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- these are based on grouping the statistics of each token to a total score.
- the score is used in making resolution on the filtering.
- According to ⁵, the token T which denote the **spamminess (spam rating)** is computed as

$$S[T] = \frac{C_{Spam}(T)}{C_{Spam}(T) + C_{Ham}(T)}$$

- Here, $C_{Spam}(T)$ = The number of spam messages containing token T , and
- $C_{Ham}(T)$ = The number of ham messages containing token T

⁵ K. Li, Z. Zhong, Fast statistical spam filter by approximate classifications, In: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, 2006. Saint Malo, France.

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- these are based on grouping the statistics of each token to a total score.
- the score is used in making resolution on the filtering.
- According to ⁵, the token T which denote the **spamminess (spam rating)** is computed as

$$S[T] = \frac{C_{Spam}(T)}{C_{Spam}(T) + C_{Ham}(T)}$$

- Here, $C_{Spam}(T)$ = The number of spam messages containing token T , and
- $C_{Ham}(T)$ = The number of ham messages containing token T
- Obviously, it is essential to merge the different token's spamminess to calculate the overall message spamminess i

⁵ K. Li, Z. Zhong, Fast statistical spam filter by approximate classifications, In: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, 2006. Saint Malo, France.

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- these are based on grouping the statistics of each token to a total score.
- the score is used in making resolution on the filtering.
- According to⁵, the token T which denote the **spamminess (spam rating)** is computed as

$$S[T] = \frac{C_{Spam}(T)}{C_{Spam}(T) + C_{Ham}(T)}$$

- Here, $C_{Spam}(T)$ = The number of spam messages containing token T , and
- $C_{Ham}(T)$ = The number of ham messages containing token T
- Obviously, it is essential to merge the different token's spamminess to calculate the overall message spamminess i
- this helps compute the probability for a message M with tokens $\{T_1, \dots, T_N\}$ to be a Spam.

⁵K. Li, Z. Zhong, Fast statistical spam filter by approximate classifications, In: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, 2006. Saint Malo, France.

ML-based Spam filtering: Naïve Bayes classifier...

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- these are based on grouping the statistics of each token to a total score.
- the score is used in making resolution on the filtering.
- According to⁵, the token T which denote the **spamminess (spam rating)** is computed as

$$S[T] = \frac{C_{Spam}(T)}{C_{Spam}(T) + C_{Ham}(T)}$$

- Here, $C_{Spam}(T)$ = The number of spam messages containing token T , and
- $C_{Ham}(T)$ = The number of ham messages containing token T
- Obviously, it is essential to merge the different token's spamminess to calculate the overall message spamminess i
- this helps compute the probability for a message M with tokens $\{T_1, \dots, T_N\}$ to be a Spam.
- Computing the product of specific token's spamminess and comparing it with the product of specific token's hamminess is essential for the purpose.

⁵K. Li, Z. Zhong, Fast statistical spam filter by approximate classifications, In: Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems, 2006. Saint Malo, France.

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Computing the product of specific token's spamminess and comparing it with the product of specific token's hamminess is essential to make classifications.

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Computing the product of specific token's spamminess and comparing it with the product of specific token's hamminess is essential to make classifications.
- this is represented by the following equation:

$$\left(H[M] = \prod_{i=1}^N (1 - S[T_i]) \right) \quad (1)$$

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Computing the product of specific token's spamminess and comparing it with the product of specific token's hamminess is essential to make classifications.
- this is represented by the following equation:

$$\left(H[M] = \prod_{i=1}^N (1 - S[T_i]) \right) \quad (1)$$

- The message is classified as Spam if the total spamminess product $S[M]$ is greater than the hamminess product $H[M]$.

How does the Naïve Bayes classifier-based ML Spam filtering work ?

- Computing the product of specific token's spamminess and comparing it with the product of specific token's hamminess is essential to make classifications.
- this is represented by the following equation:

$$\left(H[M] = \prod_{i=1}^N (1 - S[T_i]) \right) \quad (1)$$

- The message is classified as Spam if the total spamminess product $S[M]$ is greater than the hamminess product $H[M]$.
- the Naïve Bayes classification algorithm for email spam classification shown further implements the same.

ML-based Spam filtering: Naïve Bayes classifier

Algorithm 2 Naïve Bayes Classification Algorithm for Email Spam Classification

```
1: Input Email Message dataset
2: Parse each email into its component tokens
3: Compute probability for each token S [W] = Cspam(W)/(Cham(W) + Cspam(W))
4: Store spamminess values to a database
5:for each message M do
6: while (M not end) do
7: scan message for the next token Ti
8: query the database for spamminess S(Ti)
9: compute probabilities of message collected S [M] and H [M]
10: compute the total message filtering signal by: I [M] = f (S [M], H [M])
11:  $I[M] = \frac{I + S[M] - H[M]}{2}$ 
12: if I [M] > threshold then
13: msg is labeled as spam
14: else
15: msg is labeled as non-spam
16: end if
17: end while
18: end for 19: return Final Email Message Classification (Spam/Valid email)
20: end
```

Figure: Naïve Bayes classifier for Spam detection

- An example of **Swarm intelligent, Meta-heuristic, Bio-inspired** algorithm used for email Spam detection is **Firefly**.

- An example of **Swarm intelligent, Meta-heuristic, Bio-inspired** algorithm used for email Spam detection is **Firefly**.
- What is meta-heuristics ?

- An example of **Swarm intelligent, Meta-heuristic, Bio-inspired** algorithm used for email Spam detection is **Firefly**.
- What is meta-heuristics ?
 - A heuristic is a strategy that **uses information about the problem being solved** to find promising solutions.

- An example of **Swarm intelligent, Meta-heuristic, Bio-inspired** algorithm used for email Spam detection is **Firefly**.
- What is meta-heuristics ?
 - A heuristic is a strategy that **uses information about the problem being solved** to find promising solutions.
 - According to the chosen heuristic for a specific problem, the objective is not necessarily **finding the optimal solution** but only finding a **good enough solution**.

- An example of **Swarm intelligent, Meta-heuristic, Bio-inspired** algorithm used for email Spam detection is **Firefly**.
- What is meta-heuristics ?
 - A heuristic is a strategy that **uses information about the problem being solved** to find promising solutions.
 - According to the chosen heuristic for a specific problem, the objective is not necessarily **finding the optimal solution** but only finding a **good enough solution**.
 - Similar to heuristics, meta-heuristics aim to find promising results for a problem.

- An example of **Swarm intelligent, Meta-heuristic, Bio-inspired** algorithm used for email Spam detection is **Firefly**.
- What is meta-heuristics ?
 - A heuristic is a strategy that **uses information about the problem being solved** to find promising solutions.
 - According to the chosen heuristic for a specific problem, the objective is not necessarily **finding the optimal solution** but only finding a **good enough solution**.
 - Similar to heuristics, meta-heuristics aim to find promising results for a problem.
 - However, the algorithm used for a metaheuristic **is generic and can deal with different problems**. That is, the meta-heuristics replace the principle of the problem-based design with the **problem-independent design**.

- An example of **Swarm intelligent, Meta-heuristic, Bio-inspired** algorithm used for email Spam detection is **Firefly**.
- What is meta-heuristics ?
 - A heuristic is a strategy that **uses information about the problem being solved** to find promising solutions.
 - According to the chosen heuristic for a specific problem, the objective is not necessarily **finding the optimal solution** but only finding a **good enough solution**.
 - Similar to heuristics, meta-heuristics aim to find promising results for a problem.
 - However, the algorithm used for a metaheuristic **is generic and can deal with different problems**. That is, the meta-heuristics replace the principle of the problem-based design with the **problem-independent design**.
 - Famous examples of metaheuristics are **genetic algorithms, particle swarm optimization, simulated annealing, and variable neighborhood search**.

- An example of **Swarm intelligent, Meta-heuristic, Bio-inspired** algorithm used for email Spam detection is **Firefly**.
- What is meta-heuristics ?
 - A heuristic is a strategy that **uses information about the problem being solved** to find promising solutions.
 - According to the chosen heuristic for a specific problem, the objective is not necessarily **finding the optimal solution** but only finding a **good enough solution**.
 - Similar to heuristics, meta-heuristics aim to find promising results for a problem.
 - However, the algorithm used for a metaheuristic **is generic and can deal with different problems**. That is, the meta-heuristics replace the principle of the problem-based design with the **problem-independent design**.
 - Famous examples of metaheuristics are **genetic algorithms, particle swarm optimization, simulated annealing, and variable neighborhood search**.
- What is Swarm Intelligent and Bio-inspired?

- What is Swarm Intelligent and Bio-inspired?

- What is Swarm Intelligent and Bio-inspired?
 - In general, swarm intelligence algorithms are **nature-inspired algorithms** developed based on the **interactions between living organisms** such as flocks of birds, ants, and fish.

- What is Swarm Intelligent and Bio-inspired?
 - In general, swarm intelligence algorithms are **nature-inspired algorithms** developed based on the **interactions between living organisms** such as flocks of birds, ants, and fish.
 - help in the **enhancement of fitness functions** in combinatorial and numerical optimization problems by discovering different **combinations of values**.

- What is Swarm Intelligent and Bio-inspired?
 - In general, swarm intelligence algorithms are **nature-inspired algorithms** developed based on the **interactions between living organisms** such as flocks of birds, ants, and fish.
 - help in the **enhancement of fitness functions** in combinatorial and numerical optimization problems by **discovering different combinations of values**.
 - fitness function - a function **which takes a candidate solution** to the problem as input and produces as output **how “fit” or how “good”** the solution is, with respect to the problem in consideration.

- What is Swarm Intelligent and Bio-inspired?
 - In general, swarm intelligence algorithms are **nature-inspired algorithms** developed based on the **interactions between living organisms** such as flocks of birds, ants, and fish.
 - help in the **enhancement of fitness functions** in combinatorial and numerical optimization problems by **discovering different combinations of values**.
 - fitness function - a function **which takes a candidate solution** to the problem as input and produces as output **how “fit” or how “good”** the solution is, with respect to the problem in consideration.
- What is Firefly algorithm ?

- What is Swarm Intelligent and Bio-inspired?
 - In general, swarm intelligence algorithms are **nature-inspired algorithms** developed based on the **interactions between living organisms** such as flocks of birds, ants, and fish.
 - help in the **enhancement of fitness functions** in combinatorial and numerical optimization problems by **discovering different combinations of values**.
 - fitness function - a function **which takes a candidate solution** to the problem as input and produces as output **how “fit” or how “good”** the solution is, with respect to the problem in consideration.
- What is Firefly algorithm ?
- Firefly is a Swarm intelligent, Meta-heuristic, Bio-inspired algorithm, developed by Yang in 2008, inspired by the flashing behavior of fireflies.

- What is Swarm Intelligent and Bio-inspired?
 - In general, swarm intelligence algorithms are **nature-inspired algorithms** developed based on the **interactions between living organisms** such as flocks of birds, ants, and fish.
 - help in the **enhancement of fitness functions** in combinatorial and numerical optimization problems by **discovering different combinations of values**.
 - fitness function - a function **which takes a candidate solution** to the problem as input and produces as output **how “fit” or how “good”** the solution is, with respect to the problem in consideration.
- What is Firefly algorithm ?
- Firefly is a Swarm intelligent, Meta-heuristic, Bio-inspired algorithm, developed by Yang in 2008, inspired by the flashing behavior of fireflies.
- The algorithm is based on the concept of communication among fireflies.

- What is Swarm Intelligent and Bio-inspired?
 - In general, swarm intelligence algorithms are **nature-inspired algorithms** developed based on the **interactions between living organisms** such as flocks of birds, ants, and fish.
 - help in the **enhancement of fitness functions** in combinatorial and numerical optimization problems by **discovering different combinations of values**.
 - fitness function - a function **which takes a candidate solution** to the problem as input and produces as output **how “fit” or how “good”** the solution is, with respect to the problem in consideration.
- What is Firefly algorithm ?
- Firefly is a Swarm intelligent, Meta-heuristic, Bio-inspired algorithm, developed by Yang in 2008, inspired by the flashing behavior of fireflies.
- The algorithm is based on the concept of communication among fireflies.
- The population of fireflies show characteristic **luminary flashing activities** to function as **attracting the partners, communication, and risk warning** for predators.

- The fireflies normally generate a little spark with a particular format subject to what they are involved in.

- The fireflies normally generate a little spark with a particular format subject to what they are involved in.
- They **also thus share information among themselves** by means of their **sparkling** attribute.

- The fireflies normally generate a little spark with a particular format subject to what they are involved in.
- They **also thus share information among themselves** by means of their **sparkling attribute**.
- With about 2000 firefly species in the world, **each one uses a dissimilar sparkling format**.

- The fireflies normally generate a little spark with a particular format subject to what they are involved in.
- They also thus share information among themselves by means of their **sparkling** attribute.
- With about 2000 firefly species in the world, each one uses a dissimilar sparkling format.
- Depending on **the form of the light, the right companion will communicate in return by either imitating the same form or answering back by using a precise form.**

- The fireflies normally generate a little spark with a particular format subject to what they are involved in.
- They also thus share information among themselves by means of their **sparkling** attribute.
- With about 2000 firefly species in the world, each one uses a dissimilar sparkling format.
- Depending on **the form of the light, the right companion will communicate in** return by either **imitating the same form** or answering back by using a precise form.
- Conversely, the intensity of light declines owing to distance.

- The fireflies normally generate a little spark with a particular format subject to what they are involved in.
- They also thus share information among themselves by means of their **sparkling attribute**.
- With about 2000 firefly species in the world, each one uses a dissimilar sparkling format.
- Depending on **the form of the light, the right companion will communicate in return** by either **imitating the same form** or answering back by using a precise form.
- Conversely, the intensity of light declines owing to distance.
- Therefore, a sparkling light exuding from a firefly gets a response from fireflies around it within a visual range of the flash.

- The fireflies normally generate a little spark with a particular format subject to what they are involved in.
- They also thus share information among themselves by means of their **sparkling attribute**.
- With about 2000 firefly species in the world, each one uses a dissimilar sparkling format.
- Depending on **the form of the light, the right companion will communicate in return** by either **imitating the same form** or answering back by using a precise form.
- Conversely, the intensity of light declines owing to distance.
- Therefore, a sparkling light exuding from a firefly gets a response from fireflies around it within a visual range of the flash.
- In the algorithm,

- The fireflies normally generate a little spark with a particular format subject to what they are involved in.
- They also thus share information among themselves by means of their **sparkling attribute**.
- With about 2000 firefly species in the world, each one uses a dissimilar sparkling format.
- Depending on **the form of the light, the right companion will communicate in return** by either **imitating the same form** or answering back by using a precise form.
- Conversely, the intensity of light declines owing to distance.
- Therefore, a sparkling light exuding from a firefly gets a response from fireflies around it within a visual range of the flash.
- In the algorithm,
 - randomly generated solutions will be considered as fireflies, and

- The fireflies normally generate a little spark with a particular format subject to what they are involved in.
- They also thus share information among themselves by means of their **sparkling attribute**.
- With about 2000 firefly species in the world, each one uses a dissimilar sparkling format.
- Depending on **the form of the light, the right companion will communicate in return** by either **imitating the same form** or answering back by using a precise form.
- Conversely, the intensity of light declines owing to distance.
- Therefore, a sparkling light exuding from a firefly gets a response from fireflies around it within a visual range of the flash.
- In the algorithm,
 - randomly generated solutions will be considered as fireflies, and
 - brightness is assigned depending on their performance on the objective function.

- the properties of attraction and movement of fireflies could inspire an optimization algorithm in which solutions follow **better (brighter)** solutions.

Algorithm 4 Firefly Algorithm for email spam classification

```
1: Input Email corpus with M number of features  
2: Set k = 0  
3: Get population of firefly N  
4: Get the number of attributes M  
5: Initialize the firefly population  
6: for each firefly  
7: Choose the firefly which has best fitness  
8: Choose corresponding features from the testing part of the email spam corpus  
9: Test the email message  
10: k = k+1  
11: Update each firefly  
12: Classify the email message as either spam or Non-spam email  
13: end for  
14: return Final Email Message Classification (Spam/Non-spam email)  
15:end
```

- From the code its not clear whether the Firefly algorithm is used for selecting features OR for classification.
- However, lot of other work on combining Firefly (to select the features) and the Bayesian classification (to classify) for Spam detection can be found.
- That is, the traditional firefly algorithm is used to chose the optimized feature space with the best fitness.
- Once the best feature space is identified through firefly algorithm, the Spam classification is done using the naïve bayes classifier.

The Rough sets

- concept was introduced by Z Pawlak in his seminal paper of 1982.

The Rough sets

- concept was introduced by Z Pawlak in his seminal paper of 1982.
- constitute a sound basis for KDD - the organized procedure of **recognizing valid, useful, and understandable patterns** from huge and complex data sets.

The Rough sets

- concept was introduced by Z Pawlak in his seminal paper of 1982.
- constitute a sound basis for KDD - the organized procedure of **recognizing valid, useful, and understandable patterns** from huge and complex data sets.
- offer mathematical tools **to discover patterns** hidden in data.

The Rough sets

- concept was introduced by Z Pawlak in his seminal paper of 1982.
- constitute a sound basis for KDD - the organized procedure of **recognizing valid, useful, and understandable patterns** from huge and complex data sets.
- offer mathematical tools **to discover patterns** hidden in data.
- The rough sets and the fuzzy sets are complementary generalizations of classical sets.

The Rough sets

- concept was introduced by Z Pawlak in his seminal paper of 1982.
- constitute a sound basis for KDD - the organized procedure of **recognizing valid, useful, and understandable patterns** from huge and complex data sets.
- offer mathematical tools **to discover patterns** hidden in data.
- The rough sets and the fuzzy sets are complementary generalizations of classical sets.
- A rough set is an **approximation** of a crisp set by means of two sets which, respectively, represent **the lower and the upper approximation** of the original set.

The Rough sets

- concept was introduced by Z Pawlak in his seminal paper of 1982.
- constitute a sound basis for KDD - the organized procedure of **recognizing valid, useful, and understandable patterns** from huge and complex data sets.
- offer mathematical tools **to discover patterns** hidden in data.
- The rough sets and the fuzzy sets are complementary generalizations of classical sets.
- A rough set is an **approximation** of a crisp set by means of two sets which, respectively, represent **the lower and the upper approximation** of the original set.
 - The approximation sets, in general, are crisp sets, but can also be fuzzy.

The Rough sets

- concept was introduced by Z Pawlak in his seminal paper of 1982.
- constitute a sound basis for KDD - the organized procedure of **recognizing valid, useful, and understandable patterns** from huge and complex data sets.
- offer mathematical tools **to discover patterns** hidden in data.
- The rough sets and the fuzzy sets are complementary generalizations of classical sets.
- A rough set is an **approximation** of a crisp set by means of two sets which, respectively, represent **the lower and the upper approximation** of the original set.
 - The approximation sets, in general, are crisp sets, but can also be fuzzy.
- A fuzzy set in X is a collection of ordered pairs of x and $\mu(x)$, where x belongs to X (a collection of x), and $\mu(x)$, extending from 0 to 1, is the membership function of x .

The Rough sets

- concept was introduced by Z Pawlak in his seminal paper of 1982.
- constitute a sound basis for KDD - the organized procedure of **recognizing valid, useful, and understandable patterns** from huge and complex data sets.
- offer mathematical tools **to discover patterns** hidden in data.
- The rough sets and the fuzzy sets are complementary generalizations of classical sets.
- A rough set is an **approximation** of a crisp set by means of two sets which, respectively, represent **the lower and the upper approximation** of the original set.
 - The approximation sets, in general, are crisp sets, but can also be fuzzy.
- A fuzzy set in X is a collection of ordered pairs of x and $\mu(x)$, where x belongs to X (a collection of x), and $\mu(x)$, extending from 0 to 1, is the membership function of x .
- The rough set makes it possible to approximate the original crisp set by reducing it to the upper and the lower approximation.

The Rough sets

- concept was introduced by Z Pawlak in his seminal paper of 1982.
- constitute a sound basis for KDD - the organized procedure of **recognizing valid, useful, and understandable patterns** from huge and complex data sets.
- offer mathematical tools **to discover patterns** hidden in data.
- The rough sets and the fuzzy sets are complementary generalizations of classical sets.
- A rough set is an **approximation** of a crisp set by means of two sets which, respectively, represent **the lower and the upper approximation** of the original set.
 - The approximation sets, in general, are crisp sets, but can also be fuzzy.
- A fuzzy set in X is a collection of ordered pairs of x and $\mu(x)$, where x belongs to X (a collection of x), and $\mu(x)$, extending from 0 to 1, is the membership function of x .
- The rough set makes it possible to approximate the original crisp set by reducing it to the upper and the lower approximation.
- A fuzzy set, on the other hand, is a set the boundaries of which **are not sharp** (i.e. they are "fuzzy").

Rough Set Theory

- is a formal theory derived from fundamental research on logical properties of information systems.

Rough Set Theory

- is a formal theory derived from fundamental research on logical properties of information systems.
- is a new area of uncertainty mathematics closely related to fuzzy theory.

Rough Set Theory

- is a formal theory derived from fundamental research on logical properties of information systems.
- is a new area of uncertainty mathematics closely related to fuzzy theory.
- can be used

Rough Set Theory

- is a formal theory derived from fundamental research on logical properties of information systems.
- is a new area of uncertainty mathematics closely related to fuzzy theory.
- can be used
 - to discover structural relationship within imprecise and noisy data.

Rough Set Theory

- is a formal theory derived from fundamental research on logical properties of information systems.
- is a new area of uncertainty mathematics closely related to fuzzy theory.
- can be used
 - to discover structural relationship within imprecise and noisy data.
 - for the induction of (learning) approximations of concepts.

Rough Set Theory

- is a formal theory derived from fundamental research on logical properties of information systems.
- is a new area of uncertainty mathematics closely related to fuzzy theory.
- can be used
 - to discover structural relationship within imprecise and noisy data.
 - for the induction of (learning) approximations of concepts.
 - for **feature selection, feature extraction, data reduction, decision rule generation, and pattern extraction** (templates, association rules) etc.

Rough Set Theory

- is a formal theory derived from fundamental research on logical properties of information systems.
- is a new area of uncertainty mathematics closely related to fuzzy theory.
- can be used
 - to discover structural relationship within imprecise and noisy data.
 - for the induction of (learning) approximations of concepts.
 - for **feature selection, feature extraction, data reduction, decision rule generation, and pattern extraction** (templates, association rules) etc.
 - to therefore, identify **partial or total dependencies** in data, to eliminate redundant data.

Rough Set Theory

- is a formal theory derived from fundamental research on logical properties of information systems.
- is a new area of uncertainty mathematics closely related to fuzzy theory.
- can be used
 - to discover structural relationship within imprecise and noisy data.
 - for the induction of (learning) approximations of concepts.
 - for **feature selection, feature extraction, data reduction, decision rule generation, and pattern extraction** (templates, association rules) etc.
 - to therefore, identify **partial or total dependencies** in data, to eliminate redundant data.
- thus one can have an approach to deal with **null values, missing data, dynamic data** and others.

Rough set theory

- represents a suitable framework for the **automated conversion of data into knowledge.**

Rough set theory

- represents a suitable framework for the **automated conversion of data into knowledge.**
- is focused on the **breakdown of categorization of inexact, ambiguous or partial information** stated in terms of the data gotten from experience.

Rough set theory

- represents a suitable framework for the **automated conversion of data into knowledge.**
- is focused on the **breakdown of categorization of inexact, ambiguous or partial information** stated in terms of the data gotten from experience.
- is built on the hypothesis that

Rough set theory

- represents a suitable framework for the **automated conversion of data into knowledge**.
- is focused on the **breakdown of categorization of inexact, ambiguous or partial information** stated in terms of the data gotten from experience.
- is built on the hypothesis that
 - some **knowledge is associated with every object of the universe** and

Rough set theory

- represents a suitable framework for the **automated conversion of data into knowledge**.
- is focused on the **breakdown of categorization of inexact, ambiguous or partial information** stated in terms of the data gotten from experience.
- is built on the hypothesis that
 - some **knowledge is associated with every object of the universe** and
 - any inexact model can be estimated from underneath and from overhead by employing an association that is imperceptible in nature.

Rough set theory

- one of the major features is the need to discover redundancy and dependencies between features -

Rough set theory

- one of the major features is the need **to discover redundancy and dependencies between features** -
- thus can be used for classification to **discover structural relationships** within imprecise or noisy data.

Rough set theory

- one of the major features is the need **to discover redundancy and dependencies between features** -
- thus can be used for classification to **discover structural relationships** within imprecise or noisy data.
- applies to **discrete-valued attributes only** - hence continuous-valued attributes must therefore be discretized before its use.

Rough set theory

- one of the major features is the need **to discover redundancy and dependencies between features** -
- thus can be used for classification to **discover structural relationships** within imprecise or noisy data.
- applies to **discrete-valued attributes only** - hence continuous-valued attributes must therefore be discretized before its use.
- is based on the **establishment of equivalence classes** within the given training data.

Rough set theory

- one of the major features is the need to discover redundancy and dependencies between features -
- thus can be used for classification to discover structural relationships within imprecise or noisy data.
- applies to discrete-valued attributes only - hence continuous-valued attributes must therefore be discretized before its use.
- is based on the establishment of equivalence classes within the given training data.
- all the data tuples forming an equivalence class are indistinguishable, that is, the samples are identical with respect to the attributes describing the data.

Rough set theory

- one of the major features is the need to discover redundancy and dependencies between features -
- thus can be used for classification to discover structural relationships within imprecise or noisy data.
- applies to discrete-valued attributes only - hence continuous-valued attributes must therefore be discretized before its use.
- is based on the establishment of equivalence classes within the given training data.
- all the data tuples forming an equivalence class are indistinguishable, that is, the samples are identical with respect to the attributes describing the data.
- given real-world data, it is common that some classes cannot be distinguished in terms of the available attributes - rough sets can be used to approximately or "roughly" define such classes.

ML-based Spam filtering: Rough Sets & Theory Basics...

A rough set definition for a given class, C,

- is approximated by two sets - **a lower approximation** of C and **an upper approximation** of C.

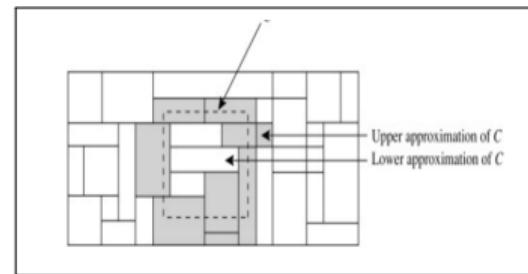


Figure: A rough set approximation of class C

ML-based Spam filtering: Rough Sets & Theory Basics...

A rough set definition for a given class, C,

- is approximated by two sets - **a lower approximation** of C and **an upper approximation** of C.
- the lower approximation of C consists of

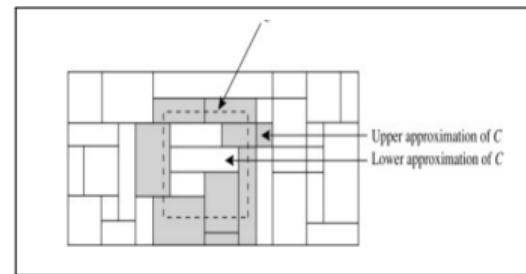


Figure: A rough set approximation of class C

ML-based Spam filtering: Rough Sets & Theory Basics...

A rough set definition for a given class, C,

- is approximated by two sets - a lower approximation of C and an upper approximation of C.
- the lower approximation of C consists of
 - all the data tuples that, based on the knowledge of the attributes, are certain to belong to C without ambiguity.

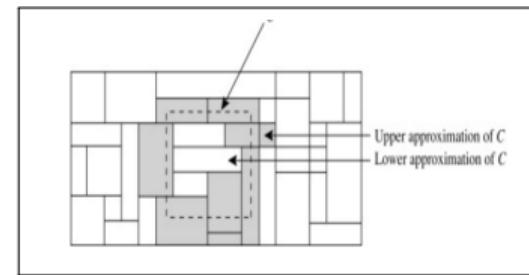


Figure: A rough set approximation of class C

ML-based Spam filtering: Rough Sets & Theory Basics...

A rough set definition for a given class, C,

- is approximated by two sets - a lower approximation of C and an upper approximation of C.
- the lower approximation of C consists of
 - all the data tuples that, based on the knowledge of the attributes, are certain to belong to C without ambiguity.
- the upper approximation of C consists of

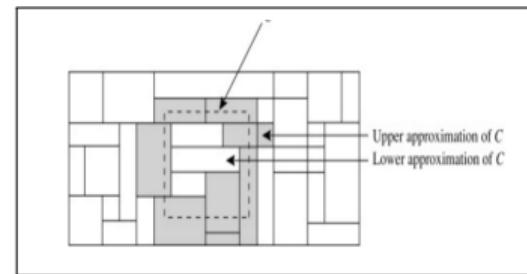


Figure: A rough set approximation of class C

ML-based Spam filtering: Rough Sets & Theory Basics...

A rough set definition for a given class, C,

- is approximated by two sets - a lower approximation of C and an upper approximation of C.
- the lower approximation of C consists of
 - all the data tuples that, based on the knowledge of the attributes, are certain to belong to C without ambiguity.
- the upper approximation of C consists of
 - all the tuples that, based on the knowledge of the attributes, cannot be described as not belonging to C.

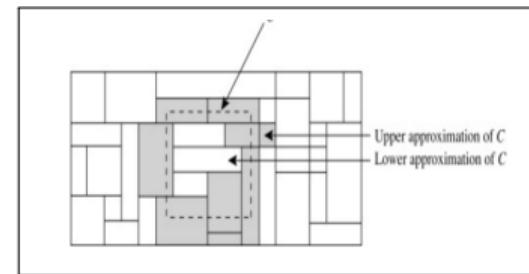


Figure: A rough set approximation of class C

ML-based Spam filtering: Rough Sets & Theory Basics...

A rough set definition for a given class, C,

- is approximated by two sets - **a lower approximation** of C and **an upper approximation** of C.
- the lower approximation of C consists of
 - all the data tuples that, based on the knowledge of the attributes, **are certain to belong to C** without ambiguity.
- the upper approximation of C consists of
 - all the tuples that, based on the knowledge of the attributes, **cannot be described as not belonging to C**.
- The lower and upper approximations for a class C are shown in Fig, where **each rectangular region represents an equivalence class**.

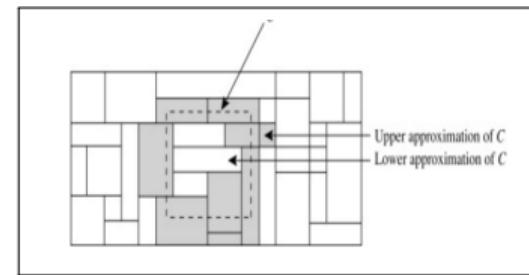


Figure: A rough set approximation of class C

ML-based Spam filtering: Rough Sets & Theory Basics...

A rough set definition for a given class, C,

- is approximated by two sets - **a lower approximation** of C and **an upper approximation** of C.
- the lower approximation of C consists of
 - all the data tuples that, based on the knowledge of the attributes, **are certain to belong to C** without ambiguity.
- the upper approximation of C consists of
 - all the tuples that, based on the knowledge of the attributes, **cannot be described as not belonging to C**.
- The lower and upper approximations for a class C are shown in Fig, where **each rectangular region represents an equivalence class**.
- Decision rules can be generated for each class.

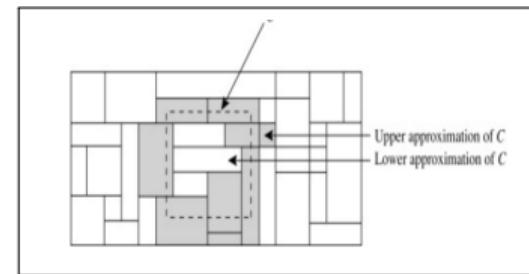


Figure: A rough set approximation of class C

- Rough sets can also be used for

- Rough sets can also be used for
 - attribute subset selection or feature reduction, where attributes that do not contribute to the classification of the given training data can be identified and removed and

- Rough sets can also be used for
 - attribute subset selection or feature reduction, where attributes that do not contribute to the classification of the given training data can be identified and removed and
 - relevance analysis where the contribution or significance of each attribute is assessed with respect to the classification task.

- Rough sets can also be used for
 - attribute subset selection or feature reduction, where attributes that do not contribute to the classification of the given training data can be identified and removed and
 - relevance analysis where the contribution or significance of each attribute is assessed with respect to the classification task.
- the problem of finding the minimal subsets (reducts) of attributes that can describe all the concepts in the given data set is NP-hard.

- Rough sets can also be used for
 - attribute subset selection or feature reduction, where attributes that do not contribute to the classification of the given training data can be identified and removed and
 - relevance analysis where the contribution or significance of each attribute is assessed with respect to the classification task.
- the problem of finding the minimal subsets (reducts) of attributes that can describe all the concepts in the given data set is NP-hard.
- however, algorithms to reduce the computational complexity have been proposed.

- Rough sets can also be used for
 - attribute subset selection or feature reduction, where attributes that do not contribute to the classification of the given training data can be identified and removed and
 - relevance analysis where the contribution or significance of each attribute is assessed with respect to the classification task.
- the problem of finding the minimal subsets (reducts) of attributes that can describe all the concepts in the given data set is NP-hard.
- however, algorithms to reduce the computational complexity have been proposed.
- in one method, for example, a discernibility matrix is used that stores the differences between attribute values for each pair of data tuples.

- Rough sets can also be used for
 - attribute subset selection or feature reduction, where attributes that do not contribute to the classification of the given training data can be identified and removed and
 - relevance analysis where the contribution or significance of each attribute is assessed with respect to the classification task.
- the problem of finding the minimal subsets (reducts) of attributes that can describe all the concepts in the given data set is NP-hard.
- however, algorithms to reduce the computational complexity have been proposed.
- in one method, for example, a discernibility matrix is used that stores the differences between attribute values for each pair of data tuples.
 - rather than searching on the entire training set, the matrix is instead searched to detect redundant attributes.

ML-based Spam filtering: Why Rough Sets for Spam filtering?

The Rough Set theory has been applied to Spam filtering because

- it provides **efficient and less time consuming algorithms** to extract hidden patterns in data.

ML-based Spam filtering: Why Rough Sets for Spam filtering?

The Rough Set theory has been applied to Spam filtering because

- it provides **efficient and less time consuming algorithms** to extract hidden patterns in data.
- also has the **capacity to identify with ease the relationships** that other conventional statistical techniques are finding **difficult to find**.

ML-based Spam filtering: Why Rough Sets for Spam filtering?

The Rough Set theory has been applied to Spam filtering because

- it provides **efficient and less time consuming algorithms** to extract hidden patterns in data.
- also has the **capacity to identify with ease the relationships** that other conventional statistical techniques are finding **difficult to find**.
 - because, it accepts the use of both quantitative and qualitative data.

ML-based Spam filtering: Why Rough Sets for Spam filtering?

The Rough Set theory has been applied to Spam filtering because

- it provides **efficient and less time consuming algorithms** to extract hidden patterns in data.
- also has the **capacity to identify with ease the relationships** that other conventional statistical techniques are finding **difficult to find**.
 - because, it accepts the use of both quantitative and qualitative data.
- has the ability to **estimate the minimum sets of data needed for grouping jobs**.

Strength of the RS classifiers for Email Spam detection

- Is they discover the importance of data and create **a group of decision rules** from the given data set.

Strength of the RS classifiers for Email Spam detection

- Is they discover the importance of data and create **a group of decision rules** from the given data set.
- It is reiterated that rough set theory **expresses imprecision by using a borderline section of a set** rather than by way of membership.

Strength of the RS classifiers for Email Spam detection

- they discover the importance of data and create **a group of decision rules** from the given data set.
- it is reiterated that rough set theory **expresses imprecision by using a borderline section of a set** rather than by way of membership.
- having the borderline section of a set

Strength of the RS classifiers for Email Spam detection

- Is they discover the importance of data and create **a group of decision rules** from the given data set.
- It is reiterated that rough set theory **expresses imprecision by using a borderline section of a set** rather than by way of membership.
- Having the borderline section of a set
 - **empty** implies that the set has **been clearly defined (exact)**,

Strength of the RS classifiers for Email Spam detection

- Is they discover the importance of data and create **a group of decision rules** from the given data set.
- It is reiterated that rough set theory **expresses imprecision by using a borderline section of a set** rather than by way of membership.
- Having the borderline section of a set
 - **empty** implies that the set has **been clearly defined (exact)**,
 - **non-empty** implies that the set is said to be **rough (inexact)**.

Strength of the RS classifiers for Email Spam detection

- Is they discover the importance of data and create **a group of decision rules** from the given data set.
- It is reiterated that rough set theory **expresses imprecision by using a borderline section of a set** rather than by way of membership.
- Having the borderline section of a set
 - **empty** implies that the set has **been clearly defined (exact)**,
 - **non-empty** implies that the set is said to be **rough (inexact)**.
- For a borderline section that contains at least one element in the set signifies that what we know about the set is not enough to exactly describe the set.

ML-based Spam filtering: Rough Set Theory-based classifier schematic

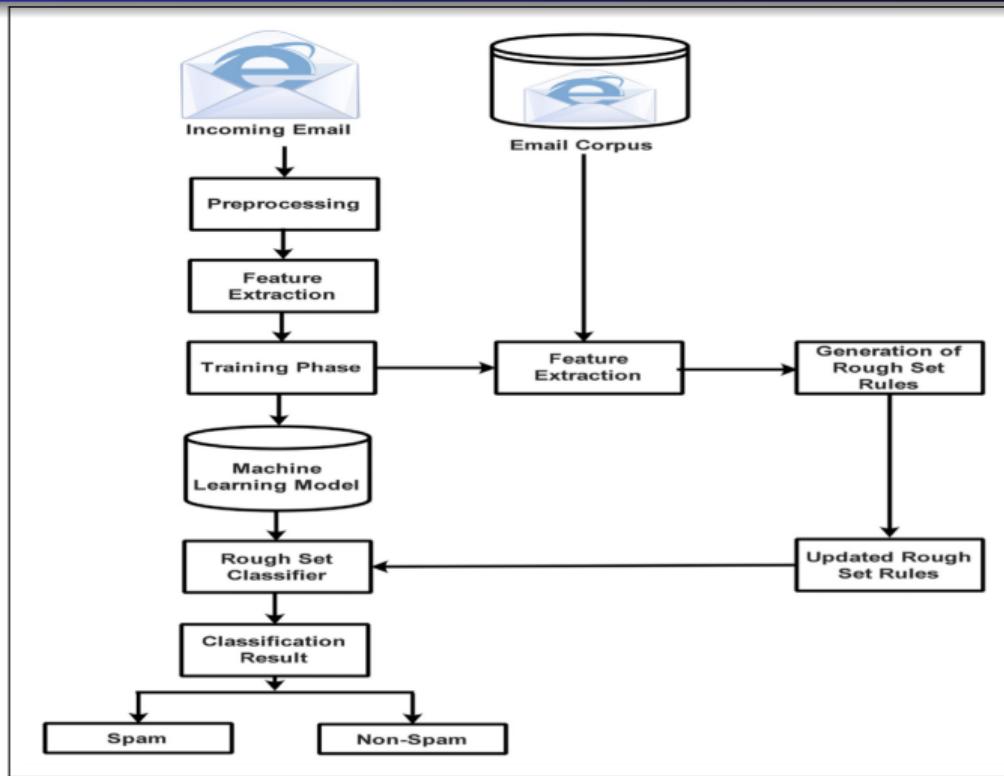


Figure: Rough Set (RS) email filtering process workflow from user mailbox.

The steps shown here are for knowledge acquisition using the rough sets

- The first step in applying the method is to **generalize the attributes using domain knowledge** to identify the **concept hierarchy**.

The steps shown here are for knowledge acquisition using the rough sets

- The first step in applying the method is to **generalize the attributes using domain knowledge** to identify the **concept hierarchy**.
- Next is to **use reduction to generate a minimal subset** of all the generalized attributes, called **a reduct**.

The steps shown here are for knowledge acquisition using the rough sets

- The first step in applying the method is to **generalize the attributes using domain knowledge** to identify the **concept hierarchy**.
- Next is to **use reduction to generate a minimal subset** of all the generalized attributes, called **a reduct**.
- A set of **general rules** may then be generated **from the reduct**

The steps shown here are for knowledge acquisition using the rough sets

- The first step in applying the method is to **generalize the attributes using domain knowledge** to identify the **concept hierarchy**.
- Next is to **use reduction to generate a minimal subset** of all the generalized attributes, called **a reduct**.
- A set of **general rules** may then be generated **from the reduct**
- The rules are such as to include all the important patterns in the data.

The steps shown here are for knowledge acquisition using the rough sets

- The first step in applying the method is to **generalize the attributes using domain knowledge** to identify the **concept hierarchy**.
- Next is to **use reduction to generate a minimal subset** of all the generalized attributes, called **a reduct**.
- A set of **general rules** may then be generated **from the reduct**
- The rules are such as to include all the important patterns in the data.
- When more than **one reduct is obtained**, one may select the best according to some criteria.

The steps shown here are for knowledge acquisition using the rough sets

- The first step in applying the method is to **generalize the attributes using domain knowledge** to identify the **concept hierarchy**.
- Next is to **use reduction to generate a minimal subset** of all the generalized attributes, called **a reduct**.
- A set of **general rules** may then be generated **from the reduct**
- The rules are such as to include all the important patterns in the data.
- When more than **one reduct is obtained**, one may select the best according to some criteria.
 - e.g. to choose the reduct that contains the smallest number of attributes.

Algorithm 5 Email spam classification algorithm using Rough Set

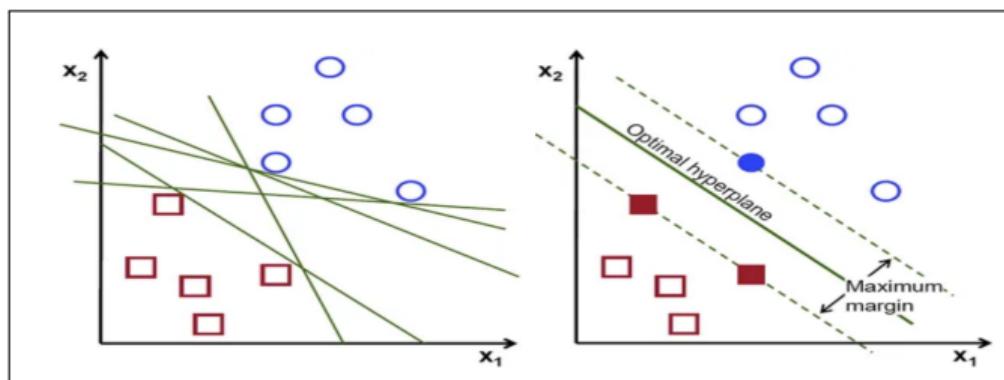
```
1: Input Email Testing Dataset (Dis_testing dataset), Rule (RUL), b
2: for  $x \in Dis\_T E$  do
3: while  $RUL(x) = 0$  do
4:   suspicious = suspicious  $\cup \{x\}$ ;
5: end while
6: Let all  $r \in RUL(x)$  cast a number in favor of the non-spam class.
7: Predict membership degree based on the decision rules;
8:  $R = r \in RUL(x) | r \text{ predicts non-spam}$ ;
9: Estimate Rel ( $Dis\_T E | x \in \text{non-spam}$ );
10:  $\text{Rel}(Dis\_T E | x \in \text{non-spam}) = \sum_{r \in R} r \text{ Predicts (non-spam)}$ ;
11:  $\text{Certainty}_x = 1/\text{cer} \times \text{Rel}(Dis\_T E | x \in \text{non-spam})$ ;
12: while  $\text{Certainty}_x \geq 1 - b$  do
13:   suspicious = suspicious  $\cup \{x\}$ ;
14: end
15: spam = spam  $\cup \{x\}$ ;
16: return Final Email Message Classification (Spam/Non-spam/Suspicious email)
17:end
```

Figure: Rough Set (RS) email filtering process pseudocode

ML-based Spam filtering: Support Vector Machine classifier

What are the SVMs?

- The purpose of the SVMs is to find a hyperplane in an N-dimensional space (N — the number of features) that distinctly classifies the data points/the features into two classes.



10

Figure: Possible Hyperplanes. SVMs to find the best

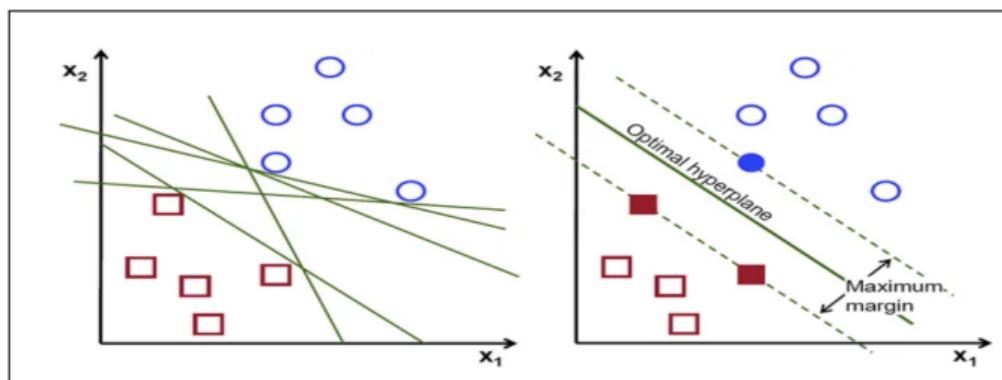
10

<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fc47>

ML-based Spam filtering: Support Vector Machine classifier

What are the SVMs?

- The purpose of the SVMs is to find **a hyperplane** in an N -dimensional space (N — the number of features) that **distinctly classifies** the data points/the features into two classes.
- SVM algorithm utilizes **many feature objects** in order to yield **a deciding line** between one class and the other.



10

Figure: Possible Hyperplanes. SVMs to find the best

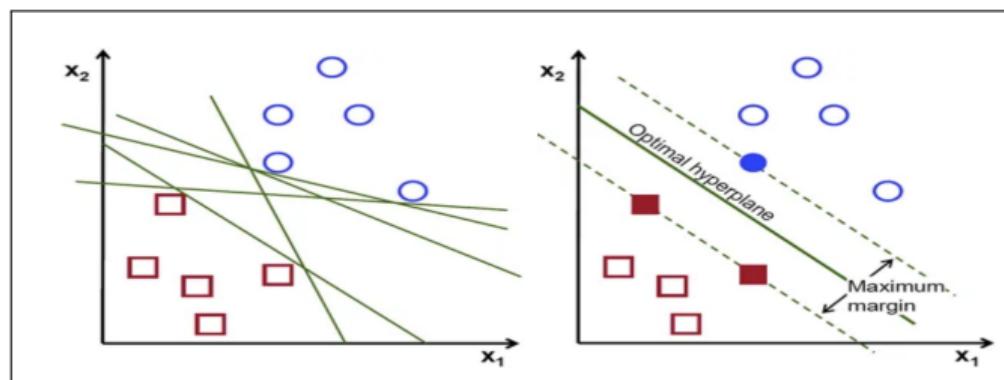
10

<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

ML-based Spam filtering: Support Vector Machine classifier

What are the SVMs?

- The purpose of the SVMs is to find a hyperplane in an N-dimensional space ($N =$ the number of features) that distinctly classifies the data points/the features into two classes.
- SVM algorithm utilizes many feature objects in order to yield a deciding line between one class and the other.
- What is a hyperplane? Why only SVM for this classification?



10

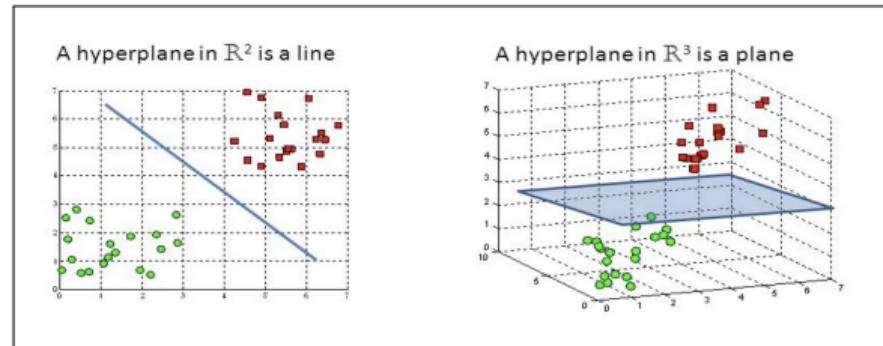
Figure: Possible Hyperplanes. SVMs to find the best

10

<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fc47>

ML-based Spam filtering: SVM: Terminologies...

- Hyperplanes are **decision boundaries** that help **classify** the data points.



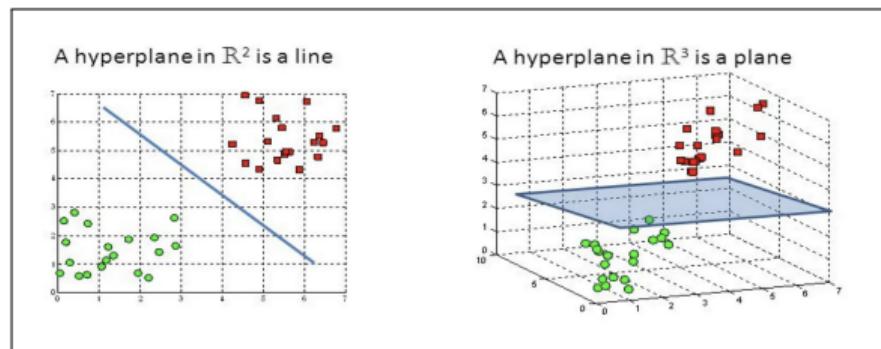
11

Figure: 2D and 3D Hyperplanes. SVMs to find the best

11 <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

ML-based Spam filtering: SVM: Terminologies...

- Hyperplanes are **decision boundaries** that help **classify** the data points.
- Data points falling on **either side of the hyperplane** can be attributed to different classes.



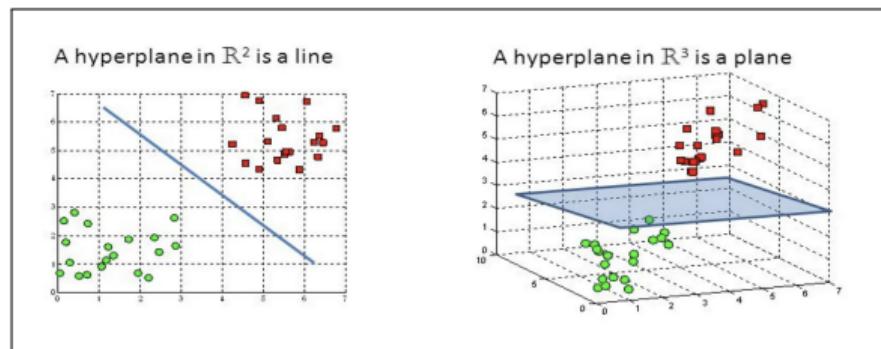
11

Figure: 2D and 3D Hyperplanes. SVMs to find the best

¹¹ <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

ML-based Spam filtering: SVM: Terminologies...

- Hyperplanes are **decision boundaries** that help **classify** the data points.
- Data points falling on **either side of the hyperplane** can be attributed to different classes.
- In addition, the **dimension** of the hyperplane depends upon **the number of features**.



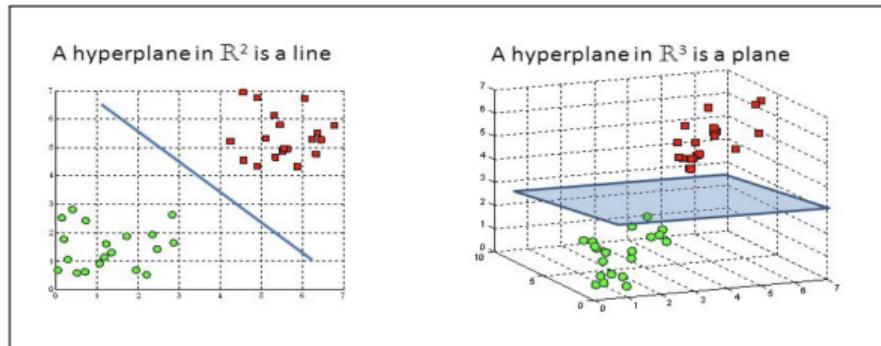
11

Figure: 2D and 3D Hyperplanes. SVMs to find the best

¹¹ <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

ML-based Spam filtering: SVM: Terminologies...

- Hyperplanes are **decision boundaries** that help **classify** the data points.
- Data points falling on **either side of the hyperplane** can be attributed to different classes.
- In addition, the **dimension** of the hyperplane depends upon **the number of features**.
 - if the number of input features is **2**, then the hyperplane is just **a line**.



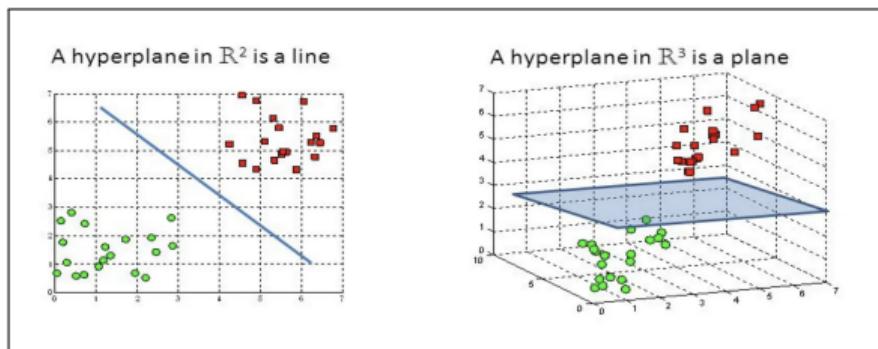
11

Figure: 2D and 3D Hyperplanes. SVMs to find the best

11 <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

ML-based Spam filtering: SVM: Terminologies...

- Hyperplanes are **decision boundaries** that help **classify** the data points.
- Data points falling on **either side of the hyperplane** can be attributed to different classes.
- In addition, the **dimension** of the hyperplane depends upon **the number of features**.
 - if the number of input features is **2**, then the hyperplane is just **a line**.
 - if the number of input features is **3**, then the hyperplane becomes a **two-dimensional plane**.



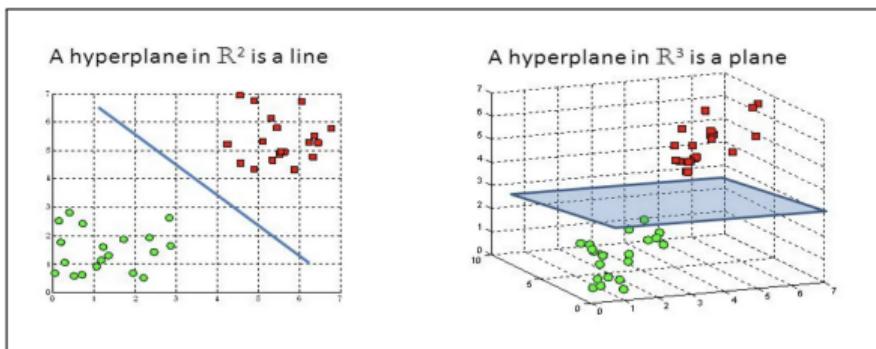
11

Figure: 2D and 3D Hyperplanes. SVMs to find the best

¹¹ <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

ML-based Spam filtering: SVM: Terminologies...

- Hyperplanes are **decision boundaries** that help **classify** the data points.
- Data points falling on **either side of the hyperplane** can be attributed to different classes.
- In addition, the **dimension** of the hyperplane depends upon **the number of features**.
 - if the number of input features is **2**, then the hyperplane is just a **line**.
 - if the number of input features is **3**, then the hyperplane becomes a **two-dimensional plane**.
 - it becomes difficult to imagine when the **number of features exceeds 3**.



11

Figure: 2D and 3D Hyperplanes. SVMs to find the best

¹¹ <https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

ML-based Spam filtering: SVM: Terminologies...

- SVM uses the notion of Support Vectors.

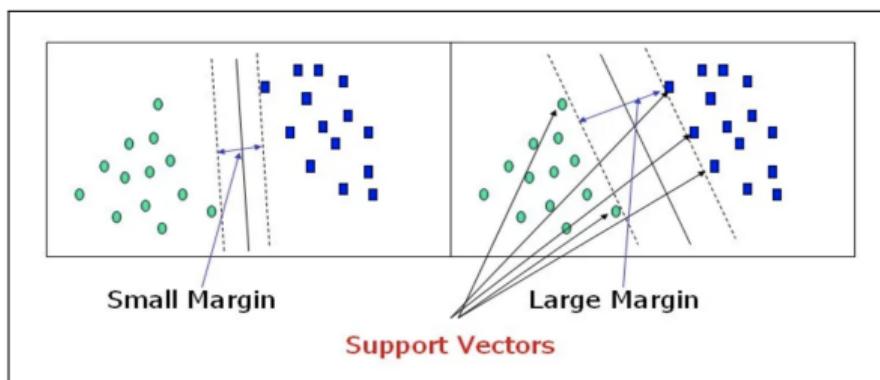
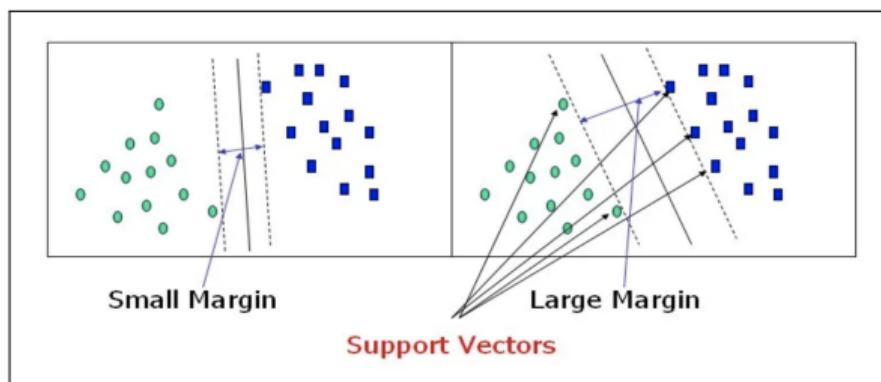


Figure: Support Vectors

ML-based Spam filtering: SVM: Terminologies...

- SVM uses the notion of Support Vectors.
- Support vectors are **data points that are closer** to the hyperplane - that influence the position and orientation of the hyperplane.



12

Figure: Support Vectors

12

<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

ML-based Spam filtering: SVM: Terminologies...

- SVM uses the notion of Support Vectors.
- Support vectors are **data points that are closer** to the hyperplane - that influence the position and orientation of the hyperplane.
- Using these support vectors, one can **maximize the margin** of the classifier.

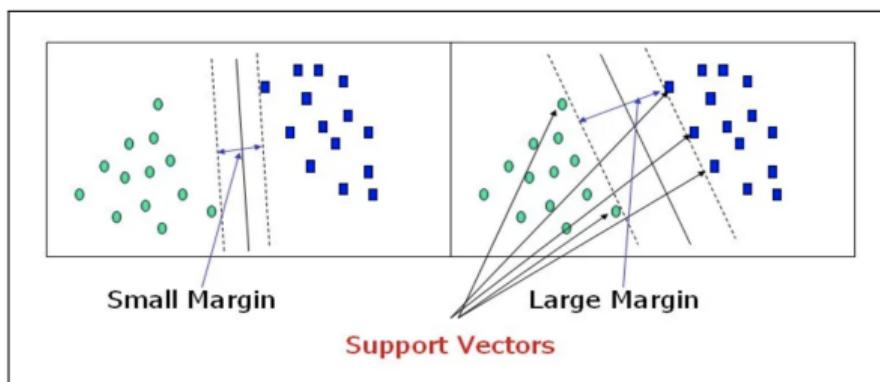


Figure: Support Vectors

ML-based Spam filtering: SVM: Terminologies...

- SVM uses the notion of Support Vectors.
- Support vectors are **data points that are closer** to the hyperplane - that influence the position and orientation of the hyperplane.
- Using these support vectors, one can **maximize the margin** of the classifier.
- Deleting the support vectors will change the position of the hyperplane.

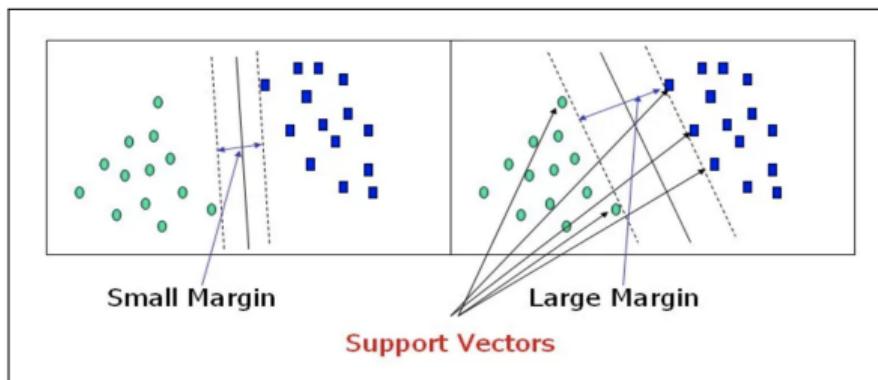


Figure: Support Vectors

ML-based Spam filtering: SVM: Terminologies...

- SVM uses the notion of Support Vectors.
- Support vectors are **data points that are closer** to the hyperplane - that influence the position and orientation of the hyperplane.
- Using these support vectors, one can **maximize the margin** of the classifier.
- Deleting the support vectors will change the position of the hyperplane.
- Support vectors are the points that help us build our SVM.

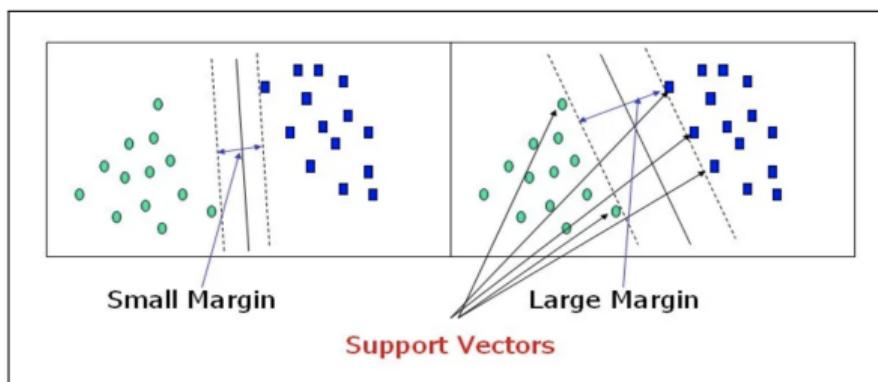


Figure: Support Vectors

ML-based Spam filtering: Why SVM for classification?

- Consider an online store that has two sets of customers, *loyal* and *disloyal*.

ML-based Spam filtering: Why SVM for classification?

- Consider an online store that has two sets of customers, *loyal* and *disloyal*.
- the goal is to determine what makes a customer loyal and disloyal with respect to **the number of orders** they place and their average order size.

ML-based Spam filtering: Why SVM for classification?

- Consider an online store that has two sets of customers, *loyal* and *disloyal*.
- the goal is to determine what makes a customer loyal and disloyal with respect to **the number of orders** they place and their average order size.
- let us assume that the data (the feature vector) appears something like this...

ML-based Spam filtering: Why SVM for classification?

- Consider an online store that has two sets of customers, *loyal* and *disloyal*.
- the goal is to determine what makes a customer loyal and disloyal with respect to **the number of orders** they place and their average order size.
- let us assume that the data (the feature vector) appears something like this...

ML-based Spam filtering: Why SVM for classification?

- Consider an online store that has two sets of customers, *loyal* and *disloyal*.
- the goal is to determine what makes a customer loyal and disloyal with respect to **the number of orders** they place and their average order size.
- let us assume that the data (the feature vector) appears something like this...

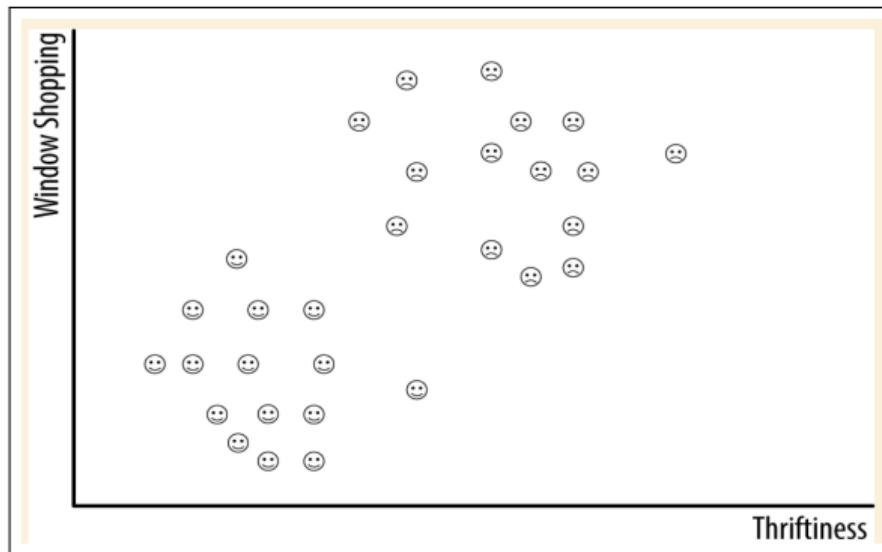


Figure: Discernible difference between the two classes

ML-based Spam filtering: Why SVM for classification?

- Consider an online store that has two sets of customers, *loyal* and *disloyal*.
- the goal is to determine what makes a customer loyal and disloyal with respect to **the number of orders** they place and their average order size.
- let us assume that the data (the feature vector) appears something like this...

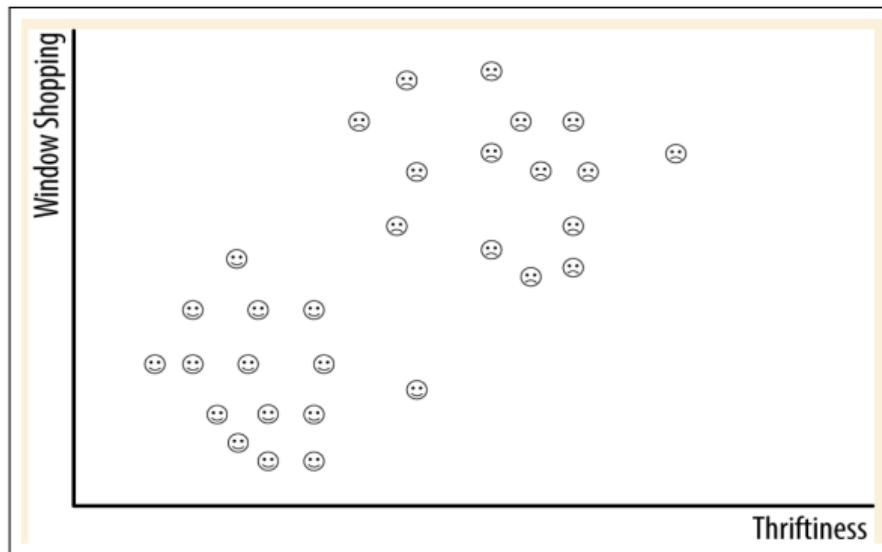


Figure: Discernible difference between the two classes

ML-based Spam filtering: Why SVM for classification?...

- If one uses a KNN classifier - one that effectively clusters things together around a centroid, the result is as shown in the figure.

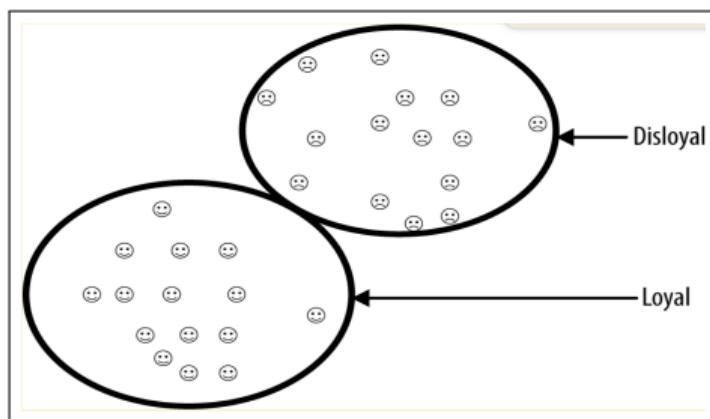


Figure: Using Centroid-based KNN Classification

ML-based Spam filtering: Why SVM for classification?...

- If one uses a KNN classifier - one that effectively clusters things together around a centroid, the result is as shown in the figure.
- Is this **the result that one wants** ?

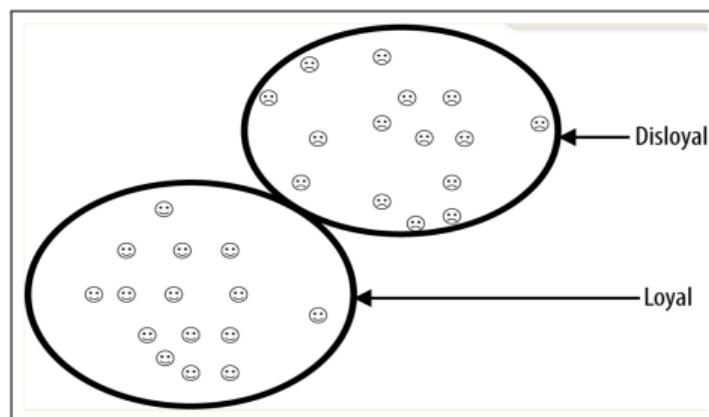


Figure: Using Centroid-based KNN Classification

ML-based Spam filtering: Why SVM for classification?...

- If one uses a KNN classifier - one that effectively clusters things together around a centroid, the result is as shown in the figure.
- Is this **the result that one wants** ?
- What is required is to find **the decision boundary** between the two.

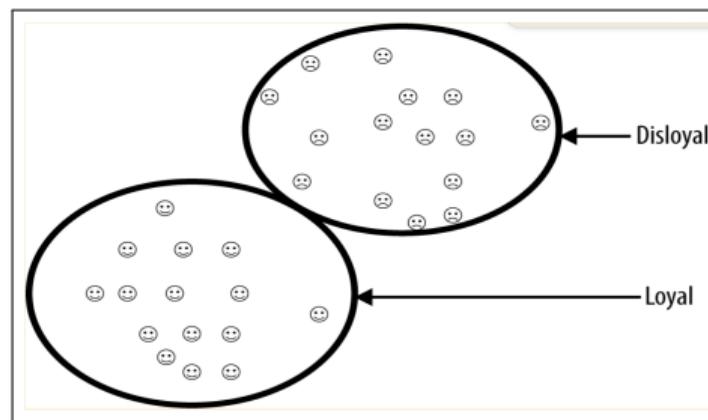


Figure: Using Centroid-based KNN Classification

ML-based Spam filtering: Why SVM for classification?...

- If one uses a KNN classifier - one that effectively clusters things together around a centroid, the result is as shown in the figure.
- Is this **the result that one wants ?**
- What is required is to find **the decision boundary** between the two.
- This decision boundary **is just a line drawn** between the two classes of data.

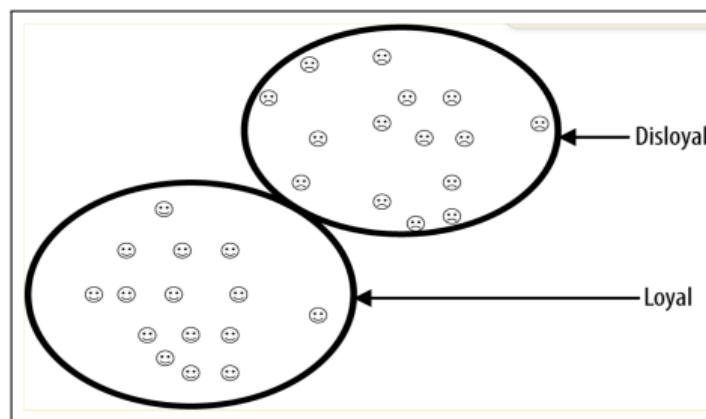


Figure: Using Centroid-based KNN Classification

ML-based Spam filtering: Why SVM for classification?...

- If one uses a KNN classifier - one that effectively clusters things together around a centroid, the result is as shown in the figure.
- Is this **the result that one wants ?**
- What is required is to find **the decision boundary** between the two.
- This decision boundary **is just a line drawn** between the two classes of data.
- The process isn't as easy as it sounds because there **COULD BE an infinite number of decision boundaries** one can draw.

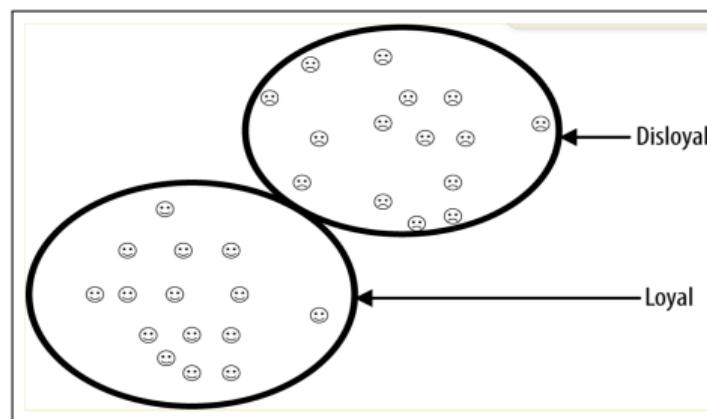


Figure: Using Centroid-based KNN Classification

ML-based Spam filtering: Why SVM for classification?...

- If one uses a KNN classifier - one that effectively clusters things together around a centroid, the result is as shown in the figure.
- Is this **the result that one wants ?**
- What is required is to find **the decision boundary** between the two.
- This decision boundary **is just a line drawn** between the two classes of data.
- The process isn't as easy as it sounds because there **COULD BE an infinite number of decision boundaries** one can draw.

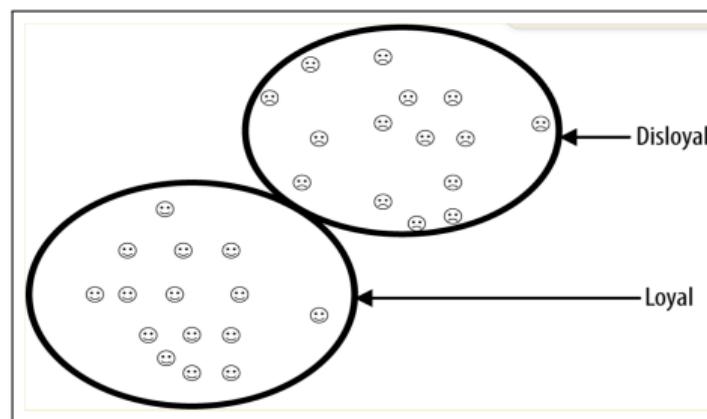


Figure: Using Centroid-based KNN Classification

ML-based Spam filtering: Why SVM for classification?...

- If one uses a KNN classifier - one that effectively clusters things together around a centroid, the result is as shown in the figure.
- Is this **the result that one wants ?**
- What is required is to find **the decision boundary** between the two.
- This decision boundary **is just a line drawn** between the two classes of data.
- The process isn't as easy as it sounds because there **COULD BE an infinite number of decision boundaries** one can draw.

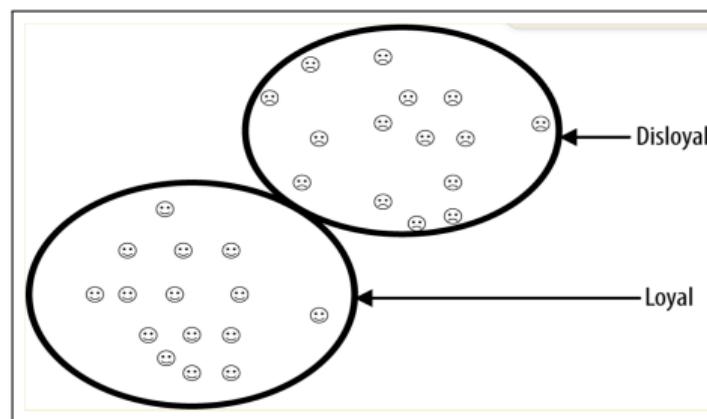


Figure: Using Centroid-based KNN Classification

- SVM exactly **serves this purpose** - that is

- SVM exactly **serves this purpose** - that is
 - originally introduced by Vladimir Vapnik in the 1980s as a way of classifying data and its modern interpretation, introduced in 1995, is **aimed to solve a two-group classification problem**.

- SVM exactly **serves this purpose** - that is
- originally introduced by Vladimir Vapnik in the 1980s as a way of classifying data and its modern interpretation, introduced in 1995, is aimed to solve a **two-group classification problem**.
- it can be used on problems that can **either be boolean (true, false), ids (3,4), or negative positive (1,-1)**.

- SVM exactly **serves this purpose** - that is
 - originally introduced by Vladimir Vapnik in the 1980s as a way of classifying data and its modern interpretation, introduced in 1995, is **aimed to solve a two-group classification problem**.
 - it can be used on problems that can **either be boolean** (true, false), **ids** (3,4), or **negative positive** (1,-1).
 - What makes SVM so special is that it **operates well in a high number of dimensions**.

- SVM exactly **serves this purpose** - that is
 - originally introduced by Vladimir Vapnik in the 1980s as a way of classifying data and its modern interpretation, introduced in 1995, is **aimed to solve a two-group classification problem**.
 - it can be used on problems that can **either be boolean** (true, false), ids (3,4), or **negative positive** (1,-1).
 - What makes SVM so special is that it **operates well in a high number of dimensions**.
 - it also avoids **the curse of dimensionality** and is also generally fast to compute.

ML-based Spam filtering: Why SVM for classification?...

- Instead of picking an arbitrary line between two sets of data, the algorithm maximizes the distance between them, as shown in the figure.

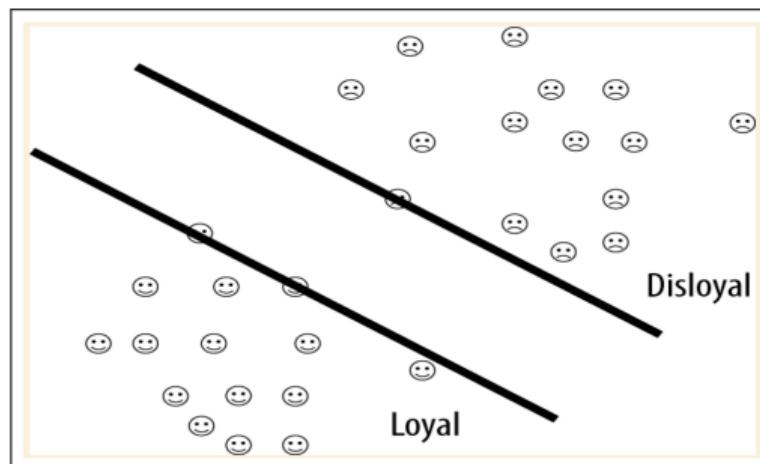


Figure: Using SVM based Classification

ML-based Spam filtering: Why SVM for classification?...

- Instead of picking an arbitrary line between two sets of data, the algorithm maximizes the distance between them, as shown in the figure.
- Hence, for instance, in the loyal versus disloyal problem, one can determine the best decision boundary between the two classes.

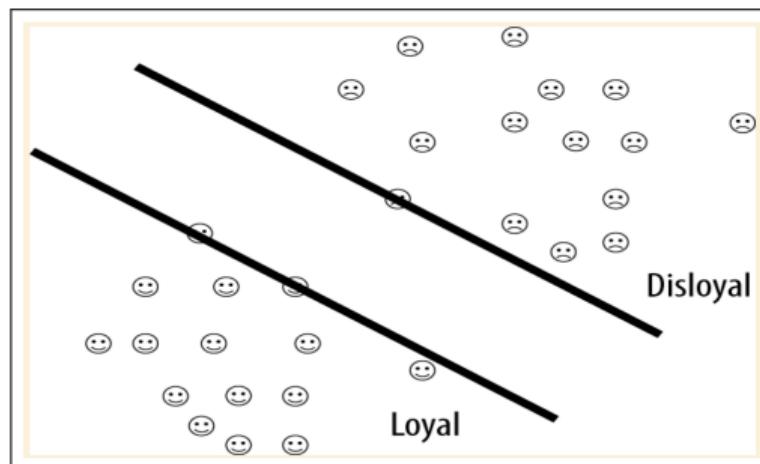


Figure: Using SVM based Classification

ML-based Spam filtering: Why SVM for classification?...

- Instead of picking an arbitrary line between two sets of data, the algorithm maximizes the distance between them, as shown in the figure.
- Hence, for instance, in the loyal versus disloyal problem, one can determine the best decision boundary between the two classes.
- Knowing that decision boundary, one could then answer the question of what makes a loyal customer versus what doesn't.

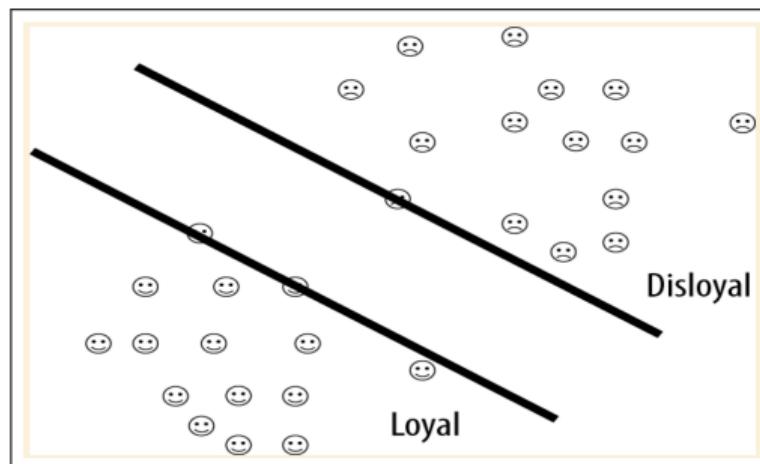


Figure: Using SVM based Classification

ML-based Spam filtering: Why SVM for classification?...

- Instead of picking an arbitrary line between two sets of data, the algorithm maximizes the distance between them, as shown in the figure.
- Hence, for instance, in the loyal versus disloyal problem, one can determine the best decision boundary between the two classes.
- Knowing that decision boundary, one could then answer the question of what makes a loyal customer versus what doesn't.

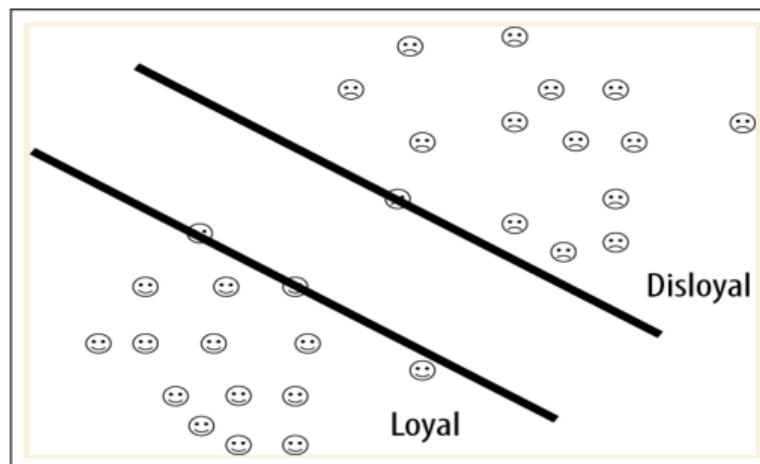
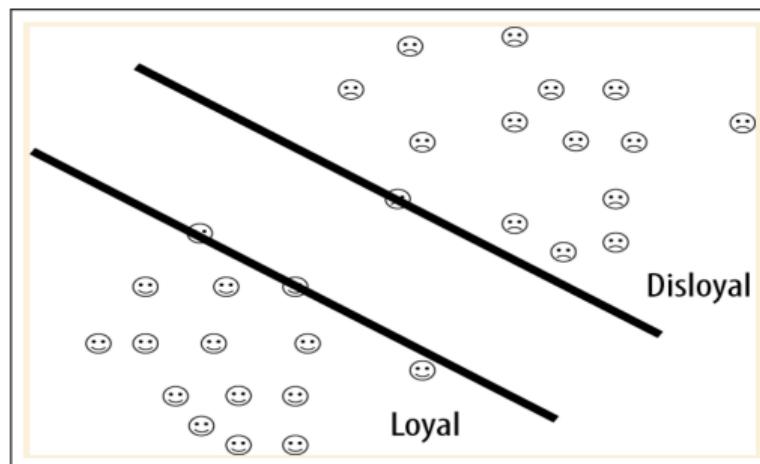


Figure: Using SVM based Classification

ML-based Spam filtering: Why SVM for classification?...

- Instead of picking an arbitrary line between two sets of data, the algorithm maximizes the distance between them, as shown in the figure.
- Hence, for instance, in the loyal versus disloyal problem, one can determine the best decision boundary between the two classes.
- Knowing that decision boundary, one could then answer the question of what makes a loyal customer versus what doesn't.



16

Figure: Using SVM based Classification

16 Thoughtful Machine Learning: A Test-Driven Approach, by Matthew Kirk, 2014, O'Reilly Media, Kindle ed

ML-based Spam filtering: SVM classification

- Thus, conceptually, SVM maximizes the distance between two sets

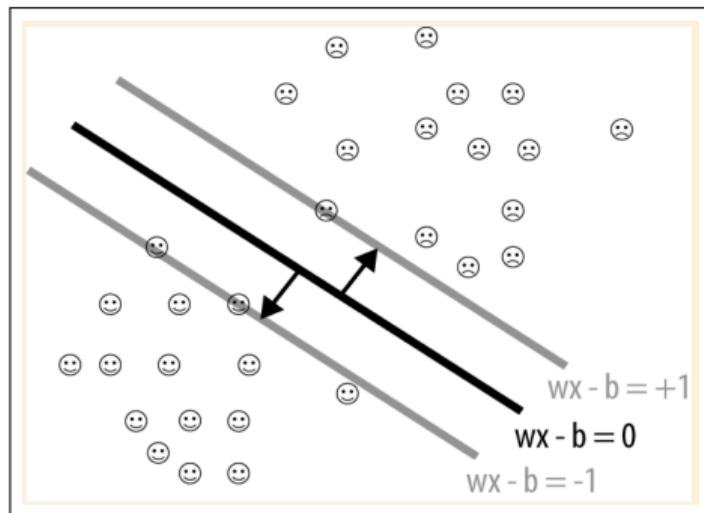


Figure: Using $wx - b = 0$ to define the hyperplane that separates the two data classes

ML-based Spam filtering: SVM classification

- Thus, conceptually, SVM maximizes the distance between two sets
 - it does so, by solving w for the function $wx - b = 0$.

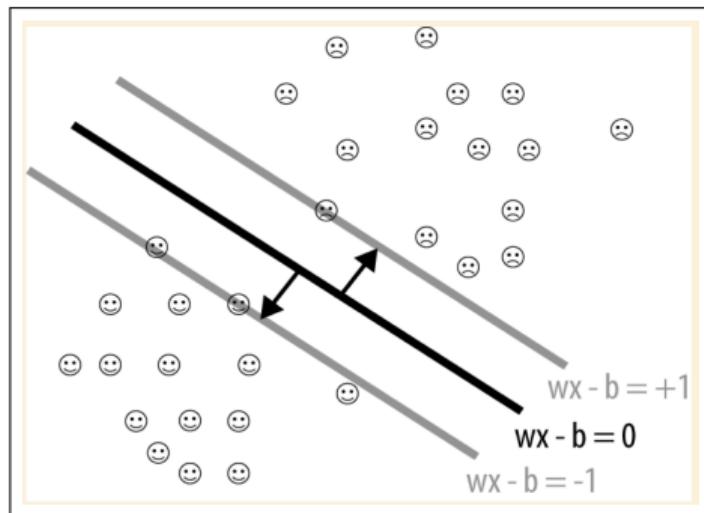


Figure: Using $wx - b = 0$ to define the hyperplane that separates the two data classes

ML-based Spam filtering: SVM classification

- Thus, conceptually, SVM **maximizes the distance** between two sets
- it does so, by solving w for the function $wx - b = 0$.
- this function could also be rewritten as $b = \sum_{i=1}^n w_i x_i$, where x_i is the value at that dimension and w_i is yet to be determined.

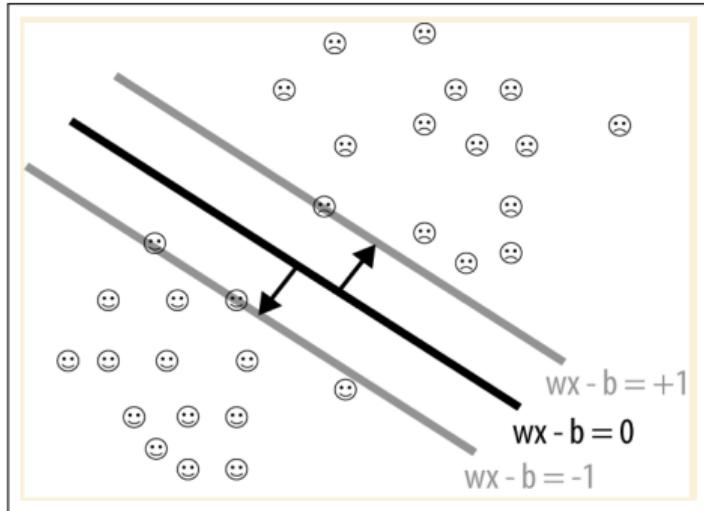


Figure: Using $wx - b = 0$ to define the hyperplane that separates the two data classes

ML-based Spam filtering: SVM classification

- Thus, conceptually, SVM **maximizes** the **distance** between two sets
- it does so, by solving w for the function $wx - b = 0$.
- this function could also be rewritten as $b = \sum_{i=1}^n w_i x_i$, where x_i is the value at that dimension and w_i is yet to be determined.
- the equations **define a hyperplane or flat surface** i.e. **n dimensional lines** between things in an **n dimensional space**.

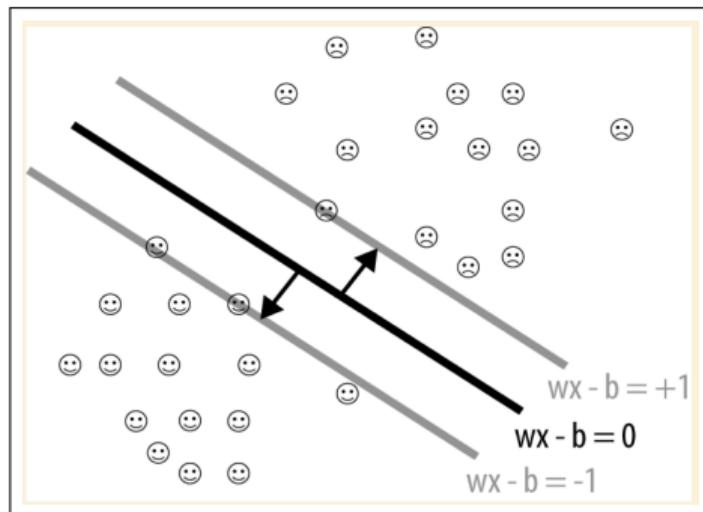


Figure: Using $wx - b = 0$ to define the hyperplane that separates the two data classes

ML-based Spam filtering: SVM classification

- Thus, conceptually, SVM **maximizes the distance** between two sets
- it does so, by solving w for the function $wx - b = 0$.
- this function could also be rewritten as $b = \sum_{i=1}^n w_i x_i$, where x_i is the value at that dimension and w_i is yet to be determined.
- the equations **define a hyperplane or flat surface** i.e. n dimensional lines between things in an n dimensional space.
- it must be emphasized that a flat surface between two sets is being drawn here to determine the distance between the most outlying data points.

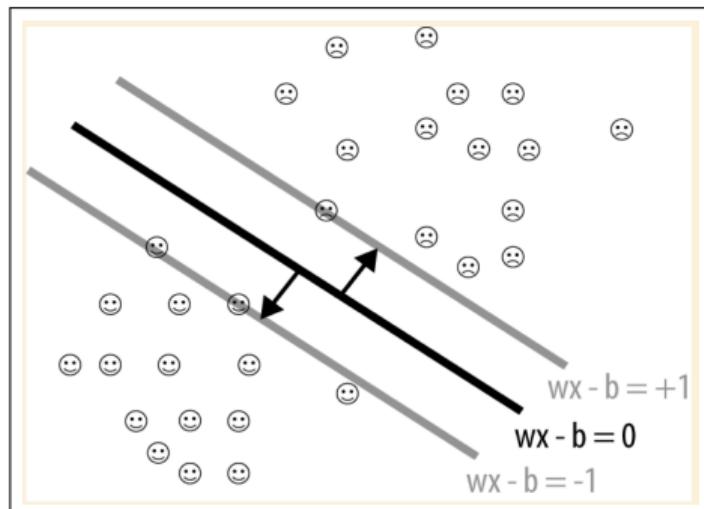


Figure: Using $wx - b = 0$ to define the hyperplane that separates the two data classes

ML-based Spam filtering: SVM classification...

- However, this is **arbitrary space** between two classes of data.

ML-based Spam filtering: SVM classification...

- However, this is **arbitrary space** between two classes of data.
- If one wants to find the maximum separation of both sets then one needs to define **two more hyperplanes**: one above and one below the existing one.

ML-based Spam filtering: SVM classification...

- However, this is **arbitrary space** between two classes of data.
- If one wants to find the maximum separation of both sets then one needs to define **two more hyperplanes: one above and one below** the existing one.
- for that one can define them as $wx + b = 1$ for the hyperplane above, and $wx + b = -1$ for the one below.

ML-based Spam filtering: SVM classification...

- However, this is **arbitrary space** between two classes of data.
- If one wants to find the maximum separation of both sets then one needs to define **two more hyperplanes: one above and one below** the existing one.
- for that one can define them as $wx + b = 1$ for the hyperplane above, and $wx + b = -1$ for the one below.
- though any actual data is not known, but there is **a margin above and a margin below**.

ML-based Spam filtering: SVM classification...

- However, this is **arbitrary space** between two classes of data.
- If one wants to find the maximum separation of both sets then one needs to define **two more hyperplanes: one above and one below** the existing one.
- for that one can define them as $wx + b = 1$ for the hyperplane above, and $wx + b = -1$ for the one below.
- though any actual data is not known, but there is **a margin above and a margin below**.
- with three hyperplanes defined, the aim is **to maximize the margin between the upper and lower hyperplane**.

- However, this is **arbitrary space** between two classes of data.
- If one wants to find the maximum separation of both sets then one needs to define **two more hyperplanes: one above and one below** the existing one.
- for that one can define them as $wx + b = 1$ for the hyperplane above, and $wx + b = -1$ for the one below.
- though any actual data is not known, but there is **a margin above and a margin below**.
- with three hyperplanes defined, the aim is **to maximize the margin between the upper and lower hyperplane**.
- this is done by having **two parallel lines** that find the general distance between.

ML-based Spam filtering: SVM classification...

- However, this is **arbitrary space** between two classes of data.
- If one wants to find the maximum separation of both sets then one needs to define **two more hyperplanes: one above and one below** the existing one.
- for that one can define them as $wx + b = 1$ for the hyperplane above, and $wx + b = -1$ for the one below.
- though any actual data is not known, but there is **a margin above and a margin below**.
- with three hyperplanes defined, the aim is **to maximize the margin between the upper and lower hyperplane**.
- this is done by having **two parallel lines** that find the general distance between.
- the only way of determining this is by **finding a hyperplane that moves perpendicular** to all the hyperplanes.

- However, this is **arbitrary space** between two classes of data.
- If one wants to find the maximum separation of both sets then one needs to define **two more hyperplanes: one above and one below** the existing one.
- for that one can define them as $wx + b = 1$ for the hyperplane above, and $wx + b = -1$ for the one below.
- though any actual data is not known, but there is **a margin above and a margin below**.
- with three hyperplanes defined, the aim is **to maximize the margin between the upper and lower hyperplane**.
- this is done by having **two parallel lines** that find the general distance between.
- the only way of determining this is by **finding a hyperplane that moves perpendicular** to all the hyperplanes.
- in this case, the perpendicular segment between the upper and lower hyperplanes happens to be $\frac{2}{\|w\|}$

- However, this is **arbitrary space** between two classes of data.
- If one wants to find the maximum separation of both sets then one needs to define **two more hyperplanes: one above and one below** the existing one.
- for that one can define them as $wx + b = 1$ for the hyperplane above, and $wx + b = -1$ for the one below.
- though any actual data is not known, but there is **a margin above and a margin below**.
- with three hyperplanes defined, the aim is **to maximize the margin between the upper and lower hyperplane**.
- this is done by having **two parallel lines** that find the general distance between.
- the only way of determining this is by **finding a hyperplane that moves perpendicular** to all the hyperplanes.
- in this case, the perpendicular segment between the upper and lower hyperplanes happens to be $\frac{2}{||w||}$
- thus instead of maximizing the margin, one can also **minimize $||w||$** , the Euclidean distance between the two hyperplanes.

Summary of SVM as a Classifier

- SVM has found applications in providing solution to quadratic programming problems that have inequality constraints and linear equality by differentiating different groups by means of a hyperplane.

Summary of SVM as a Classifier

- SVM has found applications in providing solution to quadratic programming problems that have inequality constraints and linear equality by differentiating different groups by means of a hyperplane.
- might not be as fast as other classification methods

Summary of SVM as a Classifier

- SVM has found applications in providing solution to quadratic programming problems that have inequality constraints and linear equality by differentiating different groups by means of a hyperplane.
- might not be as fast as other classification methods
- but the algorithm draws its strength from its high accuracy because of its capacity to model multidimensional borderlines that are not sequential or straightforward.

Summary of SVM as a Classifier

- SVM has found applications in providing solution to quadratic programming problems that have inequality constraints and linear equality by differentiating different groups by means of a hyperplane.
- might not be as fast as other classification methods
- but the algorithm draws its strength from its high accuracy because of its capacity to model multidimensional borderlines that are not sequential or straightforward.
- is not easily susceptible to a situation where a model is disproportionately complex such as having numerous parameters comparative to the number of observations.

Summary of SVM as a Classifier

- SVM has found applications in providing solution to quadratic programming problems that have inequality constraints and linear equality by differentiating different groups by means of a hyperplane.
- might not be as fast as other classification methods
- but the algorithm draws its strength from its high accuracy because of its capacity to model multidimensional borderlines that are not sequential or straightforward.
- is not easily susceptible to a situation where a model is disproportionately complex such as having numerous parameters comparative to the number of observations.
- is the ideal algorithm for application in the areas of digital handwriting recognition, text categorization, speaker recognition, and so on.

Summary of SVM as a Classifier

- SVM has found applications in providing solution to quadratic programming problems that have inequality constraints and linear equality by differentiating different groups by means of a hyperplane.
- might not be as fast as other classification methods
- but the algorithm draws its strength from its high accuracy because of its capacity to model multidimensional borderlines that are not sequential or straightforward.
- is not easily susceptible to a situation where a model is disproportionately complex such as having numerous parameters comparative to the number of observations.
- is the ideal algorithm for application in the areas of digital handwriting recognition, text categorization, speaker recognition, and so on.
- The SVM training and classification algorithm for Spam emails is shown on the next slide....

ML-based Spam filtering: SVM classifier pseudocode

Algorithm 6 Support Vector Machine (SVM) algorithm

```
1: Input Sample Email Message  $x$  to classify
2: A training set  $S$ , a kernel function,  $\{c_1, c_2, \dots c_{num}\}$  and  $\{\gamma_1, \gamma_2, \dots \gamma_{num}\}$ .
3: Number of nearest neighbours  $k$ .
4: for  $i = 1$  to  $num$ 
5: set  $C=C_i$ ;
6: for  $j = 1$  to  $q$ 
7: set  $\gamma=\gamma_j$ ;
8: produce a trained SVM classifier  $f(x)$  through the current merger parameter  $(C, \gamma)$ ;
9: if ( $f(x)$  is the first produced discriminant function) then
10: keep  $f(x)$  as the most ideal SVM classifier  $f^*(x)$ ;
11: else
12: compare classifier  $f(x)$  and the current best SVM classifier  $f^*(x)$  using  $k$ -fold cross-validation
13: keep classifier with a better accuracy.
14: end if
15: end for
16: end for
17: return Final Email Message Classification (Spam/Non-spam email)
18: end
```

Decision Tree

- is a **tree-structured** classifier, where **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and **each leaf node** represents the **outcome**.

Decision Tree

- is a **tree-structured** classifier, where **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and **each leaf node** represents the **outcome**.
- Two types of nodes in the tree

Decision Tree

- is a **tree-structured** classifier, where **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and **each leaf node** represents the **outcome**.
- Two types of nodes in the tree
- Decision nodes

Decision Tree

- is a **tree-structured** classifier, where **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and **each leaf node** represents the **outcome**.
- Two types of nodes in the tree
- Decision nodes
 - are used **to model any decision** and have multiple branches

Decision Tree

- is a **tree-structured** classifier, where **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and **each leaf node** represents the **outcome**.
- Two types of nodes in the tree
- Decision nodes
 - are used **to model any decision** and have multiple branches
 - indicate **certain test** to be conducted on the **value of a feature**,

Decision Tree

- is a **tree-structured** classifier, where **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and **each leaf node** represents the **outcome**.
- Two types of nodes in the tree
- Decision nodes
 - are used **to model any decision** and have multiple branches
 - indicate **certain test to be conducted** on the **value of a feature**,
 - one branch and a sub-tree (which is a subset of the larger tree) representing **every likely result** of the test

Decision Tree

- is a **tree-structured** classifier, where **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and **each leaf node** represents the **outcome**.
- Two types of nodes in the tree
- Decision nodes
 - are used **to model any decision** and have multiple branches
 - indicate **certain test to be conducted** on the **value of a feature**,
 - one branch and a sub-tree (which is a subset of the larger tree) representing **every likely result** of the test
- Leaf nodes

Decision Tree

- is a **tree-structured** classifier, where **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and **each leaf node** represents the **outcome**.
- Two types of nodes in the tree
- Decision nodes
 - are used **to model any decision** and have multiple branches
 - indicate **certain test to be conducted** on the **value of a feature**,
 - one branch and a sub-tree (which is a subset of the larger tree) representing **every likely result** of the test
- Leaf nodes
 - are **the output of those decisions** and do not contain any further branches.

Decision Tree

- is a **tree-structured** classifier, where **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and **each leaf node** represents the **outcome**.
- Two types of nodes in the tree
- Decision nodes
 - are used **to model any decision** and have multiple branches
 - indicate **certain test to be conducted** on the **value of a feature**,
 - one branch and a sub-tree (which is a subset of the larger tree) representing **every likely result** of the test
- Leaf nodes
 - are **the output of those decisions** and do not contain any further branches.
 - specify the **value of the intended feature (class)**

Decision Tree

- is a **tree-structured** classifier, where **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and **each leaf node** represents the **outcome**.
- Two types of nodes in the tree
- Decision nodes
 - are used **to model any decision** and have multiple branches
 - indicate **certain test to be conducted** on the **value of a feature**,
 - one branch and a sub-tree (which is a subset of the larger tree) representing **every likely result** of the test
- Leaf nodes
 - are **the output of those decisions** and do not contain any further branches.
 - specify the **value of the intended feature (class)**
- the decisions or the test are performed on the basis of **tangible measure of feature values** of the given dataset.

ML-based Spam filtering: Decision Tree

Decision Tree

- is a **tree-structured** classifier, where **internal nodes** represent the **features** of a dataset, **branches** represent the **decision rules** and **each leaf node** represents the **outcome**.
- Two types of nodes in the tree
- Decision nodes
 - are used **to model any decision** and have multiple branches
 - indicate **certain test to be conducted** on the **value of a feature**,
 - one branch and a sub-tree (which is a subset of the larger tree) representing **every likely result** of the test
- Leaf nodes
 - are **the output of those decisions** and do not contain any further branches.
 - specify the **value of the intended feature (class)**
- the decisions or the test are performed on the basis of **tangible measure of feature values** of the given dataset.
- thus it is a graphical representation for **getting all the possible solutions** to a problem/decision based on given conditions.

One type of DT classifier known in literature is NBTree Classifier

- a type of decision tree that **hybridizes Naïve Bayes classifier with decision tree** thereby **combining the strengths** of both algorithms.

One type of DT classifier known in literature is NBTree Classifier

- a type of decision tree that **hybridizes Naïve Bayes classifier with decision tree** thereby **combining the strengths** of both algorithms.
- works by applying **Naïve Bayes classifier at the nodes** while decision tree is developed with **one variable/feature** that is divided at each node.

ML-based Spam filtering: Decision Tree...

One type of DT classifier known in literature is NBTree Classifier

- a type of decision tree that **hybridizes Naïve Bayes classifier with decision tree** thereby **combining the strengths** of both algorithms.
- works by applying **Naïve Bayes classifier at the nodes** while decision tree is developed with **one variable/feature** that is divided at each node.
- useful for a database that is big in size.

ML-based Spam filtering: Decision Tree...

One type of DT classifier known in literature is NBTree Classifier

- a type of decision tree that **hybridizes Naïve Bayes classifier with decision tree** thereby **combining the strengths** of both algorithms.
- works by applying **Naïve Bayes classifier at the nodes** while decision tree is developed with **one variable/feature** that is divided at each node.
- useful for a database that is big in size.
- if the size of the database is **non uniform and the features are not unavoidably autonomous**, the strength of the NBTree becomes prominent.

One type of DT classifier known in literature is NBTree Classifier

- a type of decision tree that **hybridizes Naïve Bayes classifier with decision tree** thereby **combining the strengths** of both algorithms.
- works by applying **Naïve Bayes classifier at the nodes** while decision tree is developed with **one variable/feature** that is divided at each node.
- useful for a database that is big in size.
- if the size of the database is **non uniform and the features are not unavoidably autonomous**, the strength of the NBTree becomes prominent.
- the **database of the spam emails** follows the above described pattern.

One type of DT classifier known in literature is NBTree Classifier

- a type of decision tree that **hybridizes Naïve Bayes classifier with decision tree** thereby **combining the strengths** of both algorithms.
- works by applying **Naïve Bayes classifier at the nodes** while decision tree is developed with **one variable/feature** that is divided at each node.
- useful for a database that is big in size.
- if the size of the database is **non uniform and the features are not unavoidably autonomous**, the strength of the NBTree becomes prominent.
- the **database of the spam emails** follows the above described pattern.
- the task of **interpreting the classifier** is straightforward just as in Naïve Bayes.

ML-based Spam filtering: Decision Tree...

Another type of DT classifier known in literature is : viz. C4.5/J48 DT Classifier

- is a **modified**, redistributed and freely available version of **C4.5 decision tree algorithm**.

ML-based Spam filtering: Decision Tree...

Another type of DT classifier known in literature is : viz. C4.5/J48 DT Classifier

- is a **modified**, redistributed and freely available version of **C4.5 decision tree algorithm**.
- is developed by **studying data at the nodes** used to examine the **meaning of prevailing attributes**.

ML-based Spam filtering: Decision Tree...

Another type of DT classifier known in literature is : viz. C4.5/J48 DT Classifier

- is a **modified**, redistributed and freely available version of **C4.5 decision tree algorithm**.
- is developed by **studying data at the nodes** used to examine the **meaning of prevailing attributes**.
- a tree model is produced by the DT through the **use of only one feature** at a time.

ML-based Spam filtering: Decision Tree...

Another type of DT classifier known in literature is : viz. C4.5/J48 DT Classifier

- is a **modified**, redistributed and freely available version of **C4.5 decision tree algorithm**.
- is developed by **studying data at the nodes** used to examine the **meaning of prevailing attributes**.
- a tree model is produced by the DT through the **use of only one feature** at a time.
- the algorithm uses the value of **the feature** to rearrange the dataset

ML-based Spam filtering: Decision Tree...

Another type of DT classifier known in literature is : viz. C4.5/J48 DT Classifier

- is a **modified**, redistributed and freely available version of **C4.5 decision tree algorithm**.
- is developed by **studying data at the nodes** used to examine the **meaning of prevailing attributes**.
- a tree model is produced by the DT through the **use of only one feature** at a time.
- the algorithm uses the value of **the feature** to **rearrange the dataset**
- it proceeds to **search for the areas of the dataset** that obviously have **one class** and indicate those areas as leaves.

ML-based Spam filtering: Decision Tree...

Another type of DT classifier known in literature is : viz. C4.5/J48 DT Classifier

- is a **modified**, redistributed and freely available version of **C4.5 decision tree algorithm**.
- is developed by **studying data at the nodes** used to examine the **meaning of prevailing attributes**.
- a tree model is produced by the DT through the **use of only one feature** at a time.
- the algorithm uses the value of **the feature** to **rearrange the dataset**
- it proceeds to **search for the areas of the dataset** that obviously have **one class** and indicate those areas as leaves.
- for the remaining areas that contain classes that are more than one, the algorithm selects alternative features.

ML-based Spam filtering: Decision Tree...

Another type of DT classifier known in literature is : viz. C4.5/J48 DT Classifier

- is a **modified**, redistributed and freely available version of **C4.5 decision tree algorithm**.
- is developed by **studying data at the nodes** used to examine the **meaning of prevailing attributes**.
- a tree model is produced by the DT through the **use of only one feature** at a time.
- the algorithm uses the value of **the feature to rearrange the dataset**
- it proceeds to **search for the areas of the dataset** that obviously have **one class** and indicate those areas as leaves.
- for the remaining areas that contain classes that are more than one, the algorithm selects alternative features.
- also maintains the dividing process with just the number of occurrences in such areas pending the time that

ML-based Spam filtering: Decision Tree...

Another type of DT classifier known in literature is : viz. C4.5/J48 DT Classifier

- is a **modified**, redistributed and freely available version of **C4.5 decision tree algorithm**.
- is developed by **studying data at the nodes** used to examine the **meaning of prevailing attributes**.
- a tree model is produced by the DT through the **use of only one feature** at a time.
- the algorithm uses the value of **the feature** to **rearrange the dataset**
- it proceeds to **search for the areas of the dataset** that obviously have **one class** and indicate those areas as leaves.
- for the remaining areas that contain classes that are more than one, the algorithm selects alternative features.
- also maintains the dividing process with just the number of occurrences in such areas pending the time that
 - either the leaves are completely created, or

ML-based Spam filtering: Decision Tree...

Another type of DT classifier known in literature is : viz. C4.5/J48 DT Classifier

- is a **modified**, redistributed and freely available version of **C4.5 decision tree algorithm**.
- is developed by **studying data at the nodes** used to examine the **meaning of prevailing attributes**.
- a tree model is produced by the DT through the **use of only one feature** at a time.
- the algorithm uses the value of **the feature to rearrange the dataset**
- it proceeds to **search for the areas of the dataset** that obviously have **one class** and indicate those areas as leaves.
- for the remaining areas that contain classes that are more than one, the algorithm selects alternative features.
- also maintains the dividing process with just the number of occurrences in such areas pending the time that
 - either the leaves are completely created, or
 - there is absence of feature that can be utilized to create at least one leave varied in the conflicted areas.

ML-based Spam filtering: Decision Tree...

- Here, for spam detection the algorithm proposed is the one that builds the decision tree using entropy and information gain.

ML-based Spam filtering: Decision Tree...

- Here, for spam detection the algorithm proposed is the one that builds the decision tree using entropy and information gain.
- the entropy evaluates the adulteration of a random corpus of email samples

ML-based Spam filtering: Decision Tree...

- Here, for spam detection the algorithm proposed is the one that builds the decision tree using entropy and information gain.
- the entropy evaluates the adulteration of a random corpus of email samples
- the information gain is used to compute entropy by dividing the email sample by some features.

ML-based Spam filtering: Decision Tree...

- Here, for spam detection the algorithm proposed is the one that builds the decision tree using entropy and information gain.
- the entropy evaluates the adulteration of a random corpus of email samples
- the information gain is used to compute entropy by dividing the email sample by some features.
- The decision tree algorithm for classifying email messages using entropy algorithm is shown next.

ML-based Spam filtering: Decision Tree...

- Here, for spam detection the algorithm proposed is the one that builds the decision tree using entropy and information gain.
- the entropy evaluates the adulteration of a random corpus of email samples
- the information gain is used to compute entropy by dividing the email sample by some features.
- The decision tree algorithm for classifying email messages using entropy algorithm is shown next.
- But first let us try to understand the entropy and how a Decision Tree algorithm uses Entropy and Gain measures.

- What is Entropy?

- What is Entropy?
 - is the measure of **disorder** OR from other perspective it is the measure of **purity**

- What is Entropy?
 - is the measure of **disorder** OR from other perspective it is the measure of **purity**
- The Mathematical formula for Entropy is as follows, with P_i is the probability of an element/class i in the given dataset

- What is Entropy?
 - is the measure of **disorder** OR from other perspective it is the measure of **purity**
- The Mathematical formula for Entropy is as follows, with P_i is the probability of an element/class i in the given dataset

- What is Entropy?
 - is the measure of **disorder** OR from other perspective it is the measure of **purity**
- The Mathematical formula for Entropy is as follows, with P_i is the probability of an element/class i in the given dataset

$$\text{entropy } E(S) = \sum_{i=1}^{|c|} -p_i \lg p_i$$

- What is Entropy?
 - is the measure of **disorder** OR from other perspective it is the measure of **purity**
- The Mathematical formula for Entropy is as follows, with P_i is the probability of an element/class i in the given dataset

$$\text{entropy } E(S) = \sum_{i=1}^{|c|} -p_i \lg p_i$$

- Why is negative sign shown here ?

- What is Entropy?
 - is the measure of **disorder** OR from other perspective it is the measure of **purity**
- The Mathematical formula for Entropy is as follows, with P_i is the probability of an element/class i in the given dataset

$$\text{entropy } E(S) = \sum_{i=1}^{|c|} -p_i \lg p_i$$

- Why is negative sign shown here ?
- So, if we had a total of 100 data points in dataset with 30 belonging to the positive class and 70 belonging to the negative class.

- What is Entropy?
 - is the measure of **disorder** OR from other perspective it is the measure of **purity**
- The Mathematical formula for Entropy is as follows, with P_i is the probability of an element/class i in the given dataset

$$\text{entropy } E(S) = \sum_{i=1}^{|c|} -p_i \lg p_i$$

- Why is negative sign shown here ?
- So, if we had a total of 100 data points in dataset with 30 belonging to the positive class and 70 belonging to the negative class.
- Then $P_+ = 3/10$ and $P_- = 7/10$

- What is Entropy?
 - is the measure of **disorder** OR from other perspective it is the measure of **purity**
- The Mathematical formula for Entropy is as follows, with P_i is the probability of an element/class i in the given dataset

$$\text{entropy } E(S) = \sum_{i=1}^{|c|} -p_i \lg p_i$$

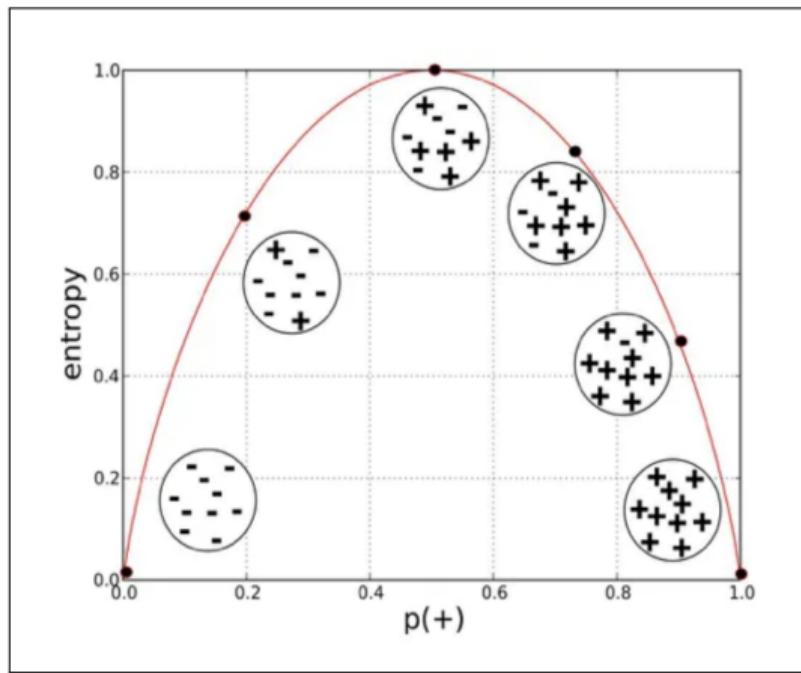
- Why is negative sign shown here ?
- So, if we had a total of 100 data points in dataset with 30 belonging to the positive class and 70 belonging to the negative class.
- Then $P_+ = 3/10$ and $P_- = 7/10$
- What would be the entropy ?

- What is Entropy?
 - is the measure of **disorder** OR from other perspective it is the measure of **purity**
- The Mathematical formula for Entropy is as follows, with P_i is the probability of an element/class i in the given dataset

$$\text{entropy } E(S) = \sum_{i=1}^{|c|} -p_i \lg p_i$$

- Why is negative sign shown here ?
- So, if we had a total of 100 data points in dataset with 30 belonging to the positive class and 70 belonging to the negative class.
- Then $P_+ = 3/10$ and $P_- = 7/10$
- What would be the entropy ?
- The value of 0.88 designates a very high level of disorder in the dataset - assuming there are two classes and the entropy ranges between 0 and 1.

What does the disorder and purity mean ?



20

Figure: Entropy: Purity and Disorder

- Entropy (E) is a measure of disorder or uncertainty and the goal of ML models designers in general is to reduce uncertainty.

- Entropy (E) is a measure of disorder or uncertainty and the goal of ML models designers in general is to reduce uncertainty.
- Thus, with Entropy value, it is known how to measure disorder.

- Entropy (E) is a measure of disorder or uncertainty and the goal of ML models designers in general is to reduce uncertainty.
- Thus, with Entropy value, it is known how to measure disorder.
- But then the question remains is how to measure the reduction of this disorder in the target variable (TV)/class, given additional information(features/independent variables) about it.

- Entropy (E) is a measure of disorder or uncertainty and the goal of ML models designers in general is to reduce uncertainty.
- Thus, with Entropy value, it is known how to measure disorder.
- But then the question remains is how to measure the reduction of this disorder in the target variable (TV)/class, given additional information(features/independent variables) about it.
- This is where Information Gain (IG) comes in - given by the equation shown.

- Entropy (E) is a measure of disorder or uncertainty and the goal of ML models designers in general is to reduce uncertainty.
- Thus, with Entropy value, it is known how to measure disorder.
- But then the question remains is how to measure the reduction of this disorder in the target variable (TV)/class, given additional information(features/independent variables) about it.
- This is where Information Gain (IG) comes in - given by the equation shown.

- Entropy (E) is a measure of disorder or uncertainty and the goal of ML models designers in general is to reduce uncertainty.
- Thus, with Entropy value, it is known how to measure disorder.
- But then the question remains is how to measure the reduction of this disorder in the target variable (TV)/class, given additional information(features/independent variables) about it.
- This is where Information Gain (IG) comes in - given by the equation shown.

$$\text{InformationGain}(Y, X) = E(Y) - E(Y/X)$$

- That is, IG is when one subtracts the entropy of Y given X from the entropy of just Y to calculate the reduction of uncertainty about Y given an additional piece of information X about Y .

- Entropy (E) is a measure of disorder or uncertainty and the goal of ML models designers in general is to reduce uncertainty.
- Thus, with Entropy value, it is known how to measure disorder.
- But then the question remains is how to measure the reduction of this disorder in the target variable (TV)/class, given additional information(features/independent variables) about it.
- This is where Information Gain (IG) comes in - given by the equation shown.

$$\text{InformationGain}(Y, X) = E(Y) - E(Y/X)$$

- That is, IG is when one subtracts the entropy of Y given X from the entropy of just Y to calculate the reduction of uncertainty about Y given an additional piece of information X about Y.
- Let us try to understand this further with the help of two examples.

How decisions trees use E and IG to decide what feature to split their nodes on as they are being trained on a data set.

Credit Rating		Liability				
		Normal	High	Total		
Excellent		3	1	4		
Good		4	2	6		
Poor		0	4	4		
Total		7	7	14		

Contingency Table

Figure: Contingency and Liability Table

How decisions trees use E and IG to decide what feature to split their nodes on as they are being trained on a data set.

$$\begin{aligned} E(\text{Liability}) &= -\frac{7}{14}\log_2\left(\frac{7}{14}\right) - \frac{7}{14}\log_2\left(\frac{7}{14}\right) \\ &= -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) \\ &= 1 \end{aligned}$$

Figure: The entropy of target variable Liability

ML-based Spam filtering: Decision Tree: Entropy and Gain Usage...

$$E(\text{Liability} \mid CR = \text{Excellent}) = -\frac{3}{4} \log_2\left(\frac{3}{4}\right) - \frac{1}{4} \log_2\left(\frac{1}{4}\right) \approx 0.811$$

$$E(\text{Liability} \mid CR = \text{Good}) = -\frac{4}{6} \log_2\left(\frac{4}{6}\right) - \frac{2}{6} \log_2\left(\frac{2}{6}\right) \approx 0.918$$

$$E(\text{Liability} \mid CR = \text{Poor}) = -0 \log_2(0) - \frac{4}{4} \log_2\left(\frac{4}{4}\right) = 0$$

Weighted Average:

$$E(\text{Liability} \mid CR) = \frac{4}{14} \times 0.811 + \frac{6}{14} \times 0.918 + \frac{4}{14} \times 0$$

$$= 0.625$$

Credit Rating		Liability		Total
		Normal	High	
Excellent	Excellent	3	1	4
	Good	4	2	6
Poor	0	4	4	4
Total	7	7	14	14

Contingency Table

Figure: The entropy of TV Liability given info about credit score

Figure: Contingency and Liability Table

Then, information gain is given by

$$IG \ (Liability, CR) = E(Liability) - E(Liability|CR)$$

$$i.e. IG \ (Liability, CR) = 1 - 0.625 = 0.375$$

Then, information gain is given by

$$IG \ (Liability, CR) = E(Liability) - E(Liability|CR)$$

$$i.e. IG \ (Liability, CR) = 1 - 0.625 = 0.375$$

- Thus, knowing the Credit Rating helped **reduce the uncertainty around the target variable**, Liability

Then, information gain is given by

$$IG \ (Liability, CR) = E(Liability) - E(Liability|CR)$$

$$i.e. IG \ (Liability, CR) = 1 - 0.625 = 0.375$$

- Thus, knowing the Credit Rating helped **reduce the uncertainty around the target variable**, Liability
- This is what a good feature is supposed to do.

Then, information gain is given by

$$IG \ (Liability, CR) = E(Liability) - E(Liability|CR)$$

$$i.e. IG \ (Liability, CR) = 1 - 0.625 = 0.375$$

- Thus, knowing the Credit Rating helped **reduce the uncertainty around the target variable**, Liability
- This is what a good feature is supposed to do.
- That is, provide information about the target variable.

Then, information gain is given by

$$IG \ (Liability, CR) = E(Liability) - E(Liability|CR)$$

$$i.e. IG \ (Liability, CR) = 1 - 0.625 = 0.375$$

- Thus, knowing the Credit Rating helped **reduce the uncertainty around the target variable**, Liability
- This is what a good feature is supposed to do.
- That is, provide information about the target variable.
- This is exactly how and why decision trees use entropy and information gain

Then, information gain is given by

$$IG \ (Liability, CR) = E(Liability) - E(Liability|CR)$$

$$i.e. IG \ (Liability, CR) = 1 - 0.625 = 0.375$$

- Thus, knowing the Credit Rating helped **reduce the uncertainty around the target variable**, Liability
- This is what a good feature is supposed to do.
- That is, provide information about the target variable.
- This is what exactly how and why decision trees use entropy and information gain
 - to determine which feature to split their nodes on

Then, information gain is given by

$$IG \ (Liability, CR) = E(Liability) - E(Liability|CR)$$

$$i.e. IG \ (Liability, CR) = 1 - 0.625 = 0.375$$

- Thus, knowing the Credit Rating helped **reduce the uncertainty around the target variable**, Liability
- This is what a good feature is supposed to do.
- That is, provide information about the target variable.
- This is what exactly how and why decision trees use entropy and information gain
 - to determine which feature to split their nodes on
 - to get closer to predicting the target variable with each split and

Then, information gain is given by

$$IG \ (Liability, CR) = E(Liability) - E(Liability|CR)$$

$$\text{i.e. } IG \ (Liability, CR) = 1 - 0.625 = 0.375$$

- Thus, knowing the Credit Rating helped **reduce the uncertainty around the target variable**, Liability
- This is what a good feature is supposed to do.
- That is, provide information about the target variable.
- This is what exactly how and why decision trees use entropy and information gain
 - to determine which feature to split their nodes on
 - to get closer to predicting the target variable with each split and
 - to determine when to stop splitting the tree!

ML-based Spam filtering: Decision Tree: Entropy and Gain Usage...

How decisions trees use E and IG to decide what feature to split their nodes on?

- Entire population consists of 30 instances.
16 belong to the **loan write-off class** and
the other 14 belong to the **loan non-write-off class**.

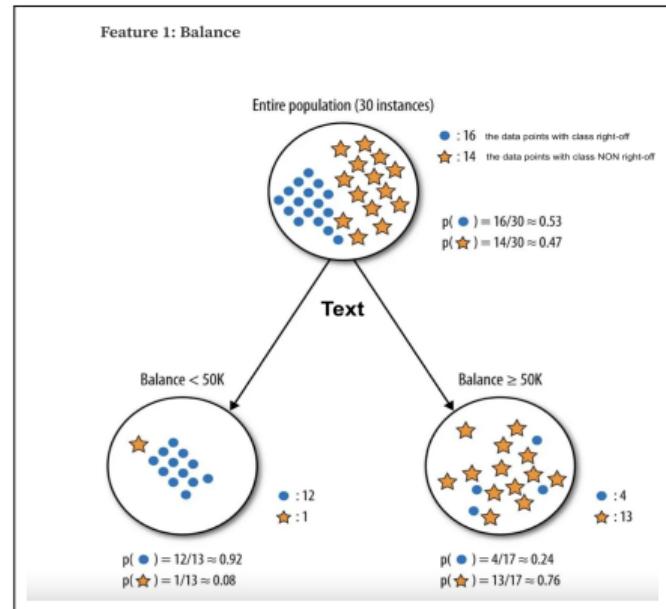


Figure: Feature1 Balance and the Information about writeoff/nonwriteoff data points

ML-based Spam filtering: Decision Tree: Entropy and Gain Usage...

How decisions trees use E and IG to decide what feature to split their nodes on?

- Entire population consists of 30 instances.
16 belong to the **loan write-off class** and
the other 14 belong to the **loan non-write-off class**.
- Two features viz. Balance and Own Residence or not.

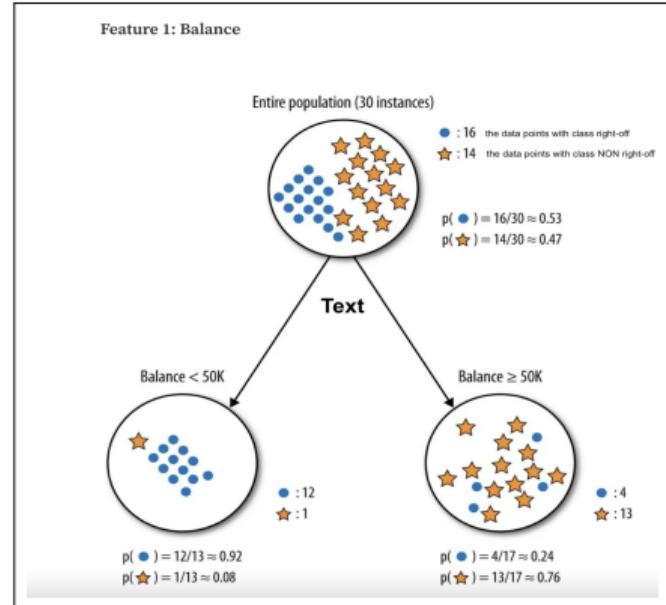


Figure: Feature1 Balance and the Information about writeoff/nonwriteoff data points

ML-based Spam filtering: Decision Tree: Entropy and Gain Usage...

How decisions trees use E and IG to decide what feature to split their nodes on?

- Entire population consists of 30 instances.
16 belong to the **loan write-off class** and
the other 14 belong to the **loan non-write-off class**.
- Two features viz. Balance and Own Residence or not.
- How does a decision tree algorithm decide what attribute to split on first ?

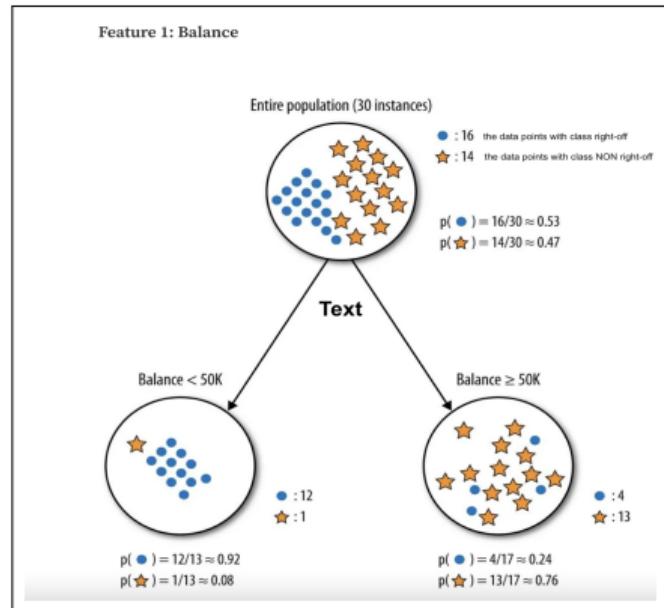


Figure: Feature1 Balance and the Information about writeoff/nonwriteoff data points

ML-based Spam filtering: Decision Tree: Entropy and Gain Usage...

How decisions trees use E and IG to decide what feature to split their nodes on?

- Entire population consists of 30 instances.
16 belong to the **loan write-off class** and
the other 14 belong to the **loan non-write-off class**.
- Two features viz. Balance and Own Residence or not.
- How does a decision tree algorithm decide what attribute to split on first ?
- What feature provides more information, OR

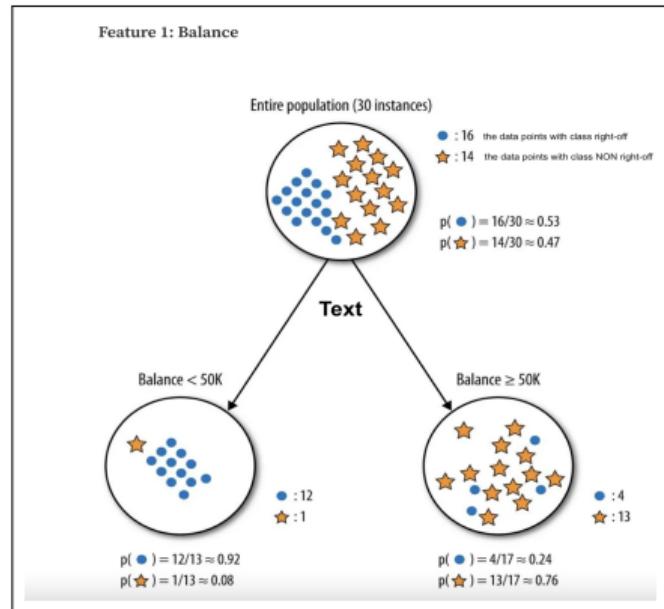


Figure: Feature1 Balance and the Information about writeoff/nonwriteoff data points

ML-based Spam filtering: Decision Tree: Entropy and Gain Usage...

How decisions trees use E and IG to decide what feature to split their nodes on?

- Entire population consists of 30 instances.
16 belong to the **loan write-off class** and
the other 14 belong to the **loan non-write-off class**.
- Two features viz. Balance and Own Residence or not.
- How does a decision tree algorithm decide what attribute to split on first ?
- What feature provides more information, OR
- What feature reduces more uncertainty about our target variable out of the two using the concepts of E and IG ?

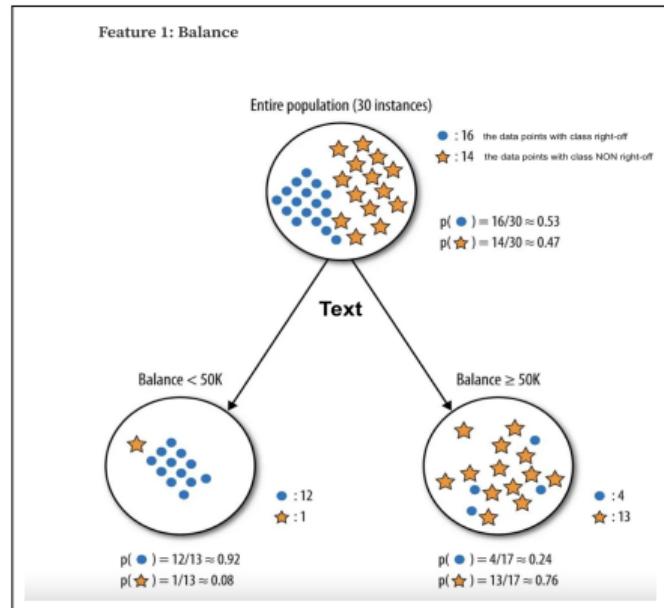


Figure: Feature1 Balance and the Information about writeoff/nonwriteoff data points

ML-based Spam filtering: Decision Tree: Entropy and Gain Usage...

How decisions trees use E and IG to decide what feature to split their nodes on?

- Entire population consists of 30 instances.
16 belong to the **loan write-off class** and
the other 14 belong to the **loan non-write-off class**.
- Two features viz. Balance and Own Residence or not.
- How does a decision tree algorithm decide what attribute to split on first ?
- What feature provides more information, OR
- What feature reduces more uncertainty about our target variable out of the two using the concepts of E and IG ?
- Let us analyze using the "dataset" in the figure....shows the feature **Balance** information

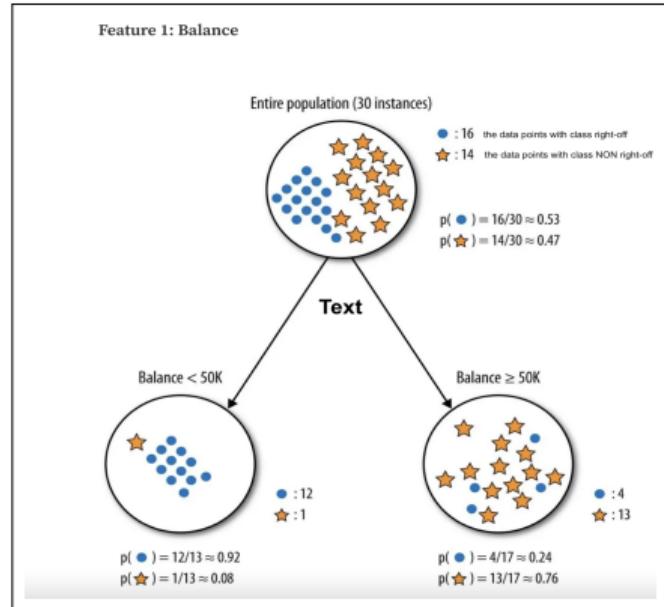


Figure: Feature1 Balance and the Information about writeoff/nonwriteoff data points

ML-based Spam filtering: Decision Tree: Entropy and Gain Usage...

$$E(\text{Parent}) = -\frac{16}{30} \log_2\left(\frac{16}{30}\right) - \frac{14}{30} \log_2\left(\frac{14}{30}\right) \approx 0.99$$

$$E(\text{Balance} < 50K) = -\frac{12}{13} \log_2\left(\frac{12}{13}\right) - \frac{1}{13} \log_2\left(\frac{1}{13}\right) \approx 0.39$$

$$E(\text{Balance} > 50K) = -\frac{4}{17} \log_2\left(\frac{4}{17}\right) - \frac{13}{17} \log_2\left(\frac{13}{17}\right) \approx 0.79$$

Weighted Average of entropy for each node:

$$\begin{aligned} E(\text{Balance}) &= \frac{13}{30} \times 0.39 + \frac{17}{30} \times 0.79 \\ &= 0.62 \end{aligned}$$

Information Gain:

$$\begin{aligned} IG(\text{Parent}, \text{Balance}) &= E(\text{Parent}) - E(\text{Balance}) \\ &= 0.99 - 0.62 \\ &= 0.37 \end{aligned}$$

Feature 1: Balance

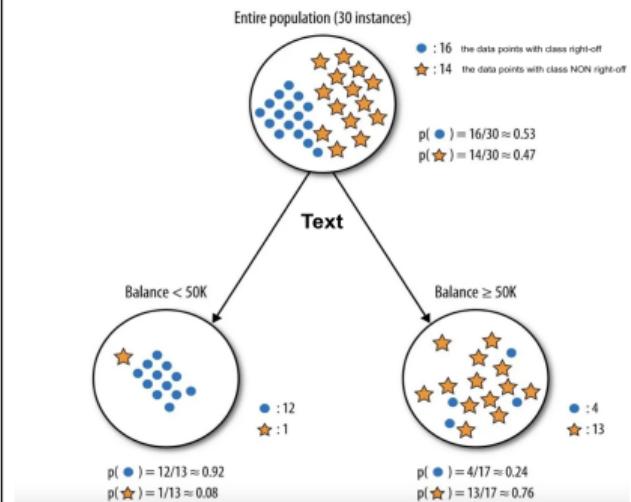


Figure: The E and IG calculations splitting on Balance

Figure: Feature 1: Balance information

ML-based Spam filtering: Decision Tree: Entropy and Gain Usage...

shows the data points for the Feature "Residence" and the E and the IG calculations

$$E(\text{Residence} = \text{OWN}) = -\frac{7}{8} \log_2\left(\frac{7}{8}\right) - \frac{1}{8} \log_2\left(\frac{1}{8}\right) \approx 0.54$$

$$E(\text{Residence} = \text{RENT}) = -\frac{4}{10} \log_2\left(\frac{4}{10}\right) - \frac{6}{10} \log_2\left(\frac{6}{10}\right) \approx 0.97$$

$$E(\text{Residence} = \text{OTHER}) = -\frac{5}{12} \log_2\left(\frac{5}{12}\right) - \frac{7}{12} \log_2\left(\frac{7}{12}\right) \approx 0.98$$

Weighted Average of entropies for each node:

$$E(\text{Residence}) = \frac{8}{30} \times 0.54 + \frac{10}{30} \times 0.97 + \frac{12}{30} \times 0.98 = 0.86$$

Information Gain:

$$\begin{aligned} IG(\text{Parent}, \text{Residence}) &= E(\text{Parent}) - E(\text{Residence}) \\ &= 0.99 - 0.86 \\ &= 0.13 \end{aligned}$$

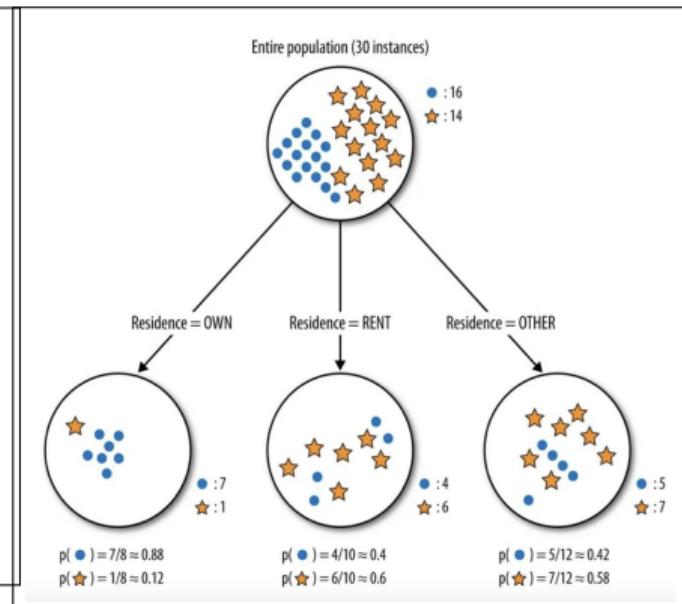


Figure: The E and IG calculations splitting on Balance

Figure: Feature 1: Balance information

ML-based Spam filtering: Decision Tree: Entropy and Gain Usage...

The child nodes from splitting on Balance do seem purer than those of Residence. However the left most node for residence is very pure but this is where the weighted avg effective

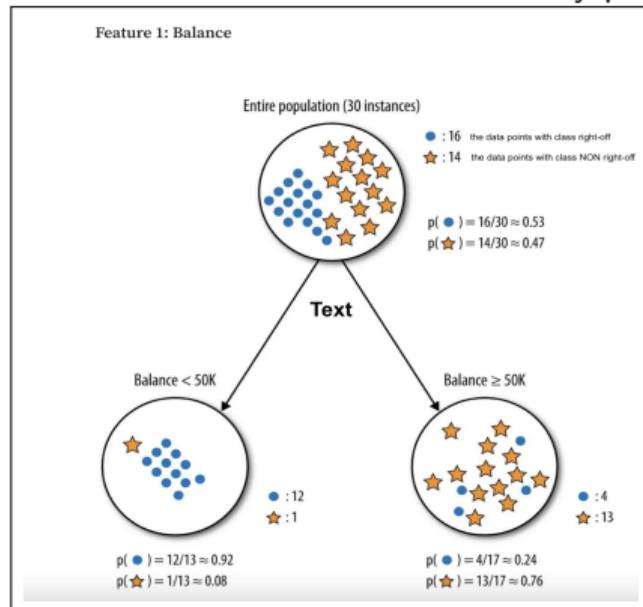


Figure: Feature1 Balance Information

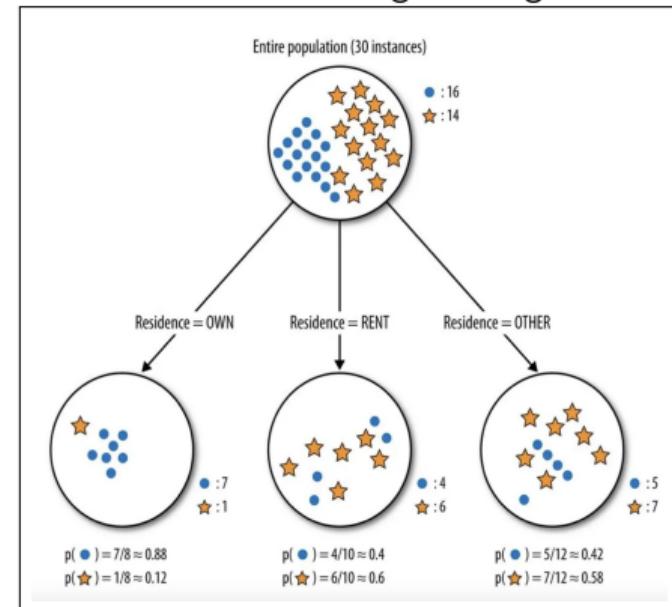


Figure: Feature2 Residence Information

ML-based Spam filtering: Decision Tree: Entropy and Gain Usage...

The $IG(Balance) = 3 * IG(Residence)$. Balance provides more information about target variable than Residence.

$$E(Parent) = -\frac{16}{30} \log_2\left(\frac{16}{30}\right) - \frac{14}{30} \log_2\left(\frac{14}{30}\right) \approx 0.99$$

$$E(Balance < 50K) = -\frac{12}{13} \log_2\left(\frac{12}{13}\right) - \frac{1}{13} \log_2\left(\frac{1}{13}\right) \approx 0.39$$

$$E(Balance > 50K) = -\frac{4}{17} \log_2\left(\frac{4}{17}\right) - \frac{13}{17} \log_2\left(\frac{13}{17}\right) \approx 0.79$$

Weighted Average of entropy for each node:

$$\begin{aligned} E(Balance) &= \frac{13}{30} \times 0.39 + \frac{17}{30} \times 0.79 \\ &= 0.62 \end{aligned}$$

Information Gain:

$$\begin{aligned} IG(Parent, Balance) &= E(Parent) - E(Balance) \\ &= 0.99 - 0.62 \\ &= 0.37 \end{aligned}$$

$$E(Residence = OWN) = -\frac{7}{8} \log_2\left(\frac{7}{8}\right) - \frac{1}{8} \log_2\left(\frac{1}{8}\right) \approx 0.54$$

$$E(Residence = RENT) = -\frac{4}{10} \log_2\left(\frac{4}{10}\right) - \frac{6}{10} \log_2\left(\frac{6}{10}\right) \approx 0.97$$

$$E(Residence = OTHER) = -\frac{5}{12} \log_2\left(\frac{5}{12}\right) - \frac{7}{12} \log_2\left(\frac{7}{12}\right) \approx 0.98$$

Weighted Average of entropies for each node:

$$E(Residence) = \frac{8}{30} \times 0.54 + \frac{10}{30} \times 0.97 + \frac{12}{30} \times 0.98 = 0.86$$

Information Gain:

$$\begin{aligned} IG(Parent, Residence) &= E(Parent) - E(Residence) \\ &= 0.99 - 0.86 \\ &= 0.13 \end{aligned}$$

Figure: Feature1 Balance E and IG Calc

Feature Balance reduces more disorder in our target variable

29

ML-based Spam filtering: Decision Tree...

- Assuming we have an email dataset E with classifications c_j , entropy is computed using

$$\text{entropy}(E) = \sum_{j=1}^{|c|} P_r(c_j) \lg P_r(c_j)$$

ML-based Spam filtering: Decision Tree...

- Assuming we have an email dataset E with classifications c_j , entropy is computed using

$$\text{entropy}(E) = \sum_{j=1}^{|c|} P_r(c_j) \lg P_r(c_j)$$

- The relationship between the entropy and information gain represented as shown below is used where $\text{entropy } A_i(E)$ is the estimated entropy of feature F_i which is exploited in dividing the email messages as either spam or legitimate mail.

$$\text{gain}(E, F_i) = \text{entropy}(D) - \text{entropy}_{F_i}(E)$$

ML-based Spam filtering: Decision Tree...

- Assuming we have an email dataset E with classifications c_j , entropy is computed using

$$\text{entropy}(E) = \sum_{j=1}^{|c|} P_r(c_j) \lg P_r(c_j)$$

- The relationship between the entropy and information gain represented as shown below is used where $\text{entropy } A_i(E)$ is the estimated entropy of feature F_i which is exploited in dividing the email messages as either spam or legitimate mail.

$$\text{gain}(E, F_i) = \text{entropy}(D) - \text{entropy}_{F_i}(E)$$

- The decision tree algorithm for classifying email messages using entropy algorithm is shown next.

Algorithm 7 Decision Tree algorithm for Spam Filtering

- 1: **Input** Email Message dataset
- 2: Compute entropy for dataset
- 3: **while** condition **do**
- 4: **for** every attribute/feature
- 5: calculate entropy for all categorical values
- 6: take average information entropy for the current attribute.
- 7: calculate gain for the current attribute
- 8: pick the highest gain attribute
- 9: **end for**
- 10: **end while**
- 11: **return** Final Email Message Classification (Spam/Non-spam email)
- 12:**end**

Revisiting Categories of Anomaly-based Spam Detection Mechanisms

Anomaly-based Spam Detection: Fundamental Approaches

Anomaly-based filtering of email Spam can be categorized into:

- Conventional Machine Learning-based approaches

Anomaly-based Spam Detection: Fundamental Approaches

Anomaly-based filtering of email Spam can be categorized into:

- Conventional Machine Learning-based approaches
- Deep Learning-based approaches

Anomaly-based Spam Detection: Fundamental Approaches

Anomaly-based filtering of email Spam can be categorized into:

- Conventional Machine Learning-based approaches
- Deep Learning-based approaches

Anomaly-based Spam Detection: Fundamental Approaches

Anomaly-based filtering of email Spam can be categorized into:

- Conventional Machine Learning-based approaches
- Deep Learning-based approaches

Conventional Machine Learning-based approaches

- does not require **pre-defined rules** - as in case of conventional Knowledge-based approach - but instead, it requires messages which have been **pre-classified** successfully.

Anomaly-based Spam Detection: Fundamental Approaches

Anomaly-based filtering of email Spam can be categorized into:

- Conventional Machine Learning-based approaches
- Deep Learning-based approaches

Conventional Machine Learning-based approaches

- does not require **pre-defined rules** - as in case of conventional Knowledge-based approach - but instead, it requires messages which have been **pre-classified** successfully.
- such messages allow **sample messages to construct the training dataset** used to fit the model's unique learning algorithm.

Anomaly-based Spam Detection: Fundamental Approaches

Anomaly-based filtering of email Spam can be categorized into:

- Conventional Machine Learning-based approaches
- Deep Learning-based approaches

Conventional Machine Learning-based approaches

- does not require **pre-defined rules** - as in case of conventional Knowledge-based approach - but instead, it requires messages which have been **pre-classified** successfully.
- such messages allow **sample messages to construct the training dataset** used to fit the model's unique learning algorithm.
- hence, the classification of Spam emails **can adopt an ML approach for classification** for learning from the input data and a program uses the learning to classify new observations.

Anomaly-based Spam Detection: Fundamental Approaches

Anomaly-based filtering of email Spam can be categorized into:

- Conventional Machine Learning-based approaches
- Deep Learning-based approaches

Conventional Machine Learning-based approaches

- does not require **pre-defined rules** - as in case of conventional Knowledge-based approach - but instead, it requires messages which have been **pre-classified** successfully.
- such messages allow **sample messages to construct the training dataset** used to fit the model's unique learning algorithm.
- hence, the classification of Spam emails **can adopt an ML approach for classification** for learning from the input data and a program uses the learning to classify new observations.
- algorithms such as Support Vector Machines and Naïve Bayes have been investigated on their effectiveness to successfully detect and filter Spam emails

Anomaly-based Spam Detection: Fundamental Approaches

Anomaly-based filtering of email Spam can be categorized into:

- Conventional Machine Learning-based approaches
- Deep Learning-based approaches

Conventional Machine Learning-based approaches

- does not require **pre-defined rules** - as in case of conventional Knowledge-based approach - but instead, it requires messages which have been **pre-classified** successfully.
- such messages allow **sample messages to construct the training dataset** used to fit the model's unique learning algorithm.
- hence, the classification of Spam emails **can adopt an ML approach for classification** for learning from the input data and a program uses the learning to classify new observations.
- algorithms such as Support Vector Machines and Naïve Bayes have been investigated on their effectiveness to successfully detect and filter Spam emails
- drawback is it requires **feature extraction** and the associated processes.

Spam Detection background: Deep Learning-based Approaches...

Filtering of Email Spam:Deep Learning-based approach

- ML approaches have constraints, viz. the most challenging one is handling new problems that need **a vast amount of labeled data** - the solution is to identify the **feature of data input before applying** the ML algorithm.

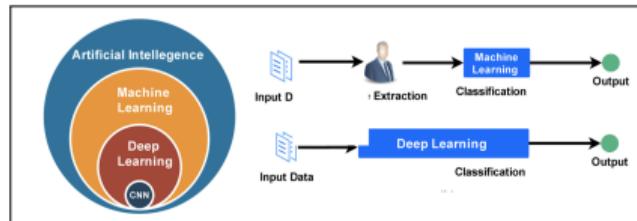


Figure: ML and DL based approach

¹ Alabadi, Montdher, and Yuksel Celik. "Anomaly detection for cyber-security based on convolution neural network: A survey." 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, 2020.

Spam Detection background: Deep Learning-based Approaches...

Filtering of Email Spam:Deep Learning-based approach

- ML approaches have constraints, viz. the most challenging one is handling new problems that need **a vast amount of labeled data** - the solution is to identify the **feature of data input before applying** the ML algorithm.
- however, this solution leads to a bottleneck because there is a limit in terms of **identifying the data features**.

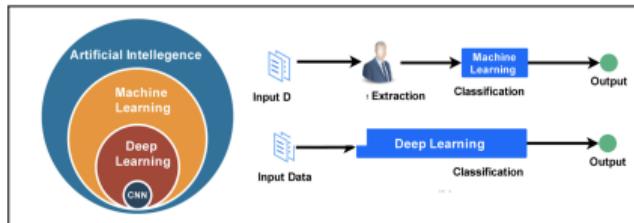


Figure: ML and DL based approach

¹ Alabadi, Montdher, and Yuksel Celik. "Anomaly detection for cyber-security based on convolution neural network: A survey." 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, 2020.

Spam Detection background: Deep Learning-based Approaches...

Filtering of Email Spam:Deep Learning-based approach

- ML approaches have constraints, viz. the most challenging one is handling new problems that need **a vast amount of labeled data** - the solution is to identify the **feature of data input before applying** the ML algorithm.
- however, this solution leads to a bottleneck because there is a limit in terms of **identifying the data features**.
- DL algorithms overcome this bottleneck - its ability to **identify the features and representation from raw data** itself.

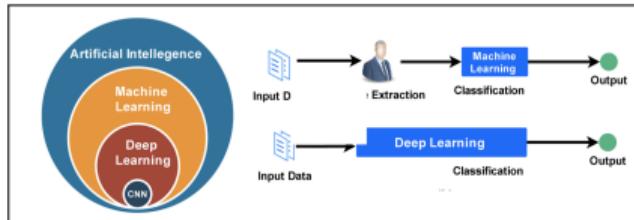


Figure: ML and DL based approach

¹ Alabadi, Montdher, and Yuksel Celik. "Anomaly detection for cyber-security based on convolution neural network: A survey." 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, 2020.

Spam Detection background: Deep Learning-based Approaches...

Filtering of Email Spam:Deep Learning-based approach

- ML approaches have constraints, viz. the most challenging one is handling new problems that need **a vast amount of labeled data** - the solution is to identify the **feature of data input before applying** the ML algorithm.
- however, this solution leads to a bottleneck because there is a limit in terms of **identifying the data features**.
- DL algorithms overcome this bottleneck - its ability to **identify the features and representation from raw data** itself.
 - using a hierarchy of distinguishing features and the ability to learn from the raw data itself.

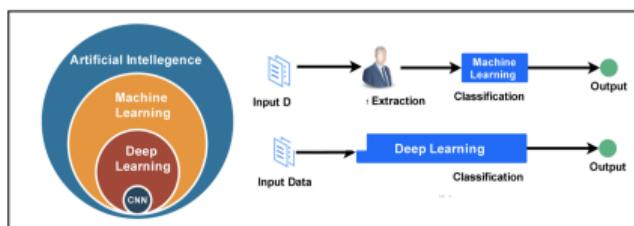


Figure: ML and DL based approach

¹ Alabadi, Montdher, and Yuksel Celik. "Anomaly detection for cyber-security based on convolution neural network: A survey." 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA). IEEE, 2020.

*Next presentation / Neural
Networks & Deep
Learning/CNN-based Spam
filtering*

Blank Slide

Blank Slide

Blank Slide

Blank Slide

Blank Slide