Text as Data: Computational Text Analysis

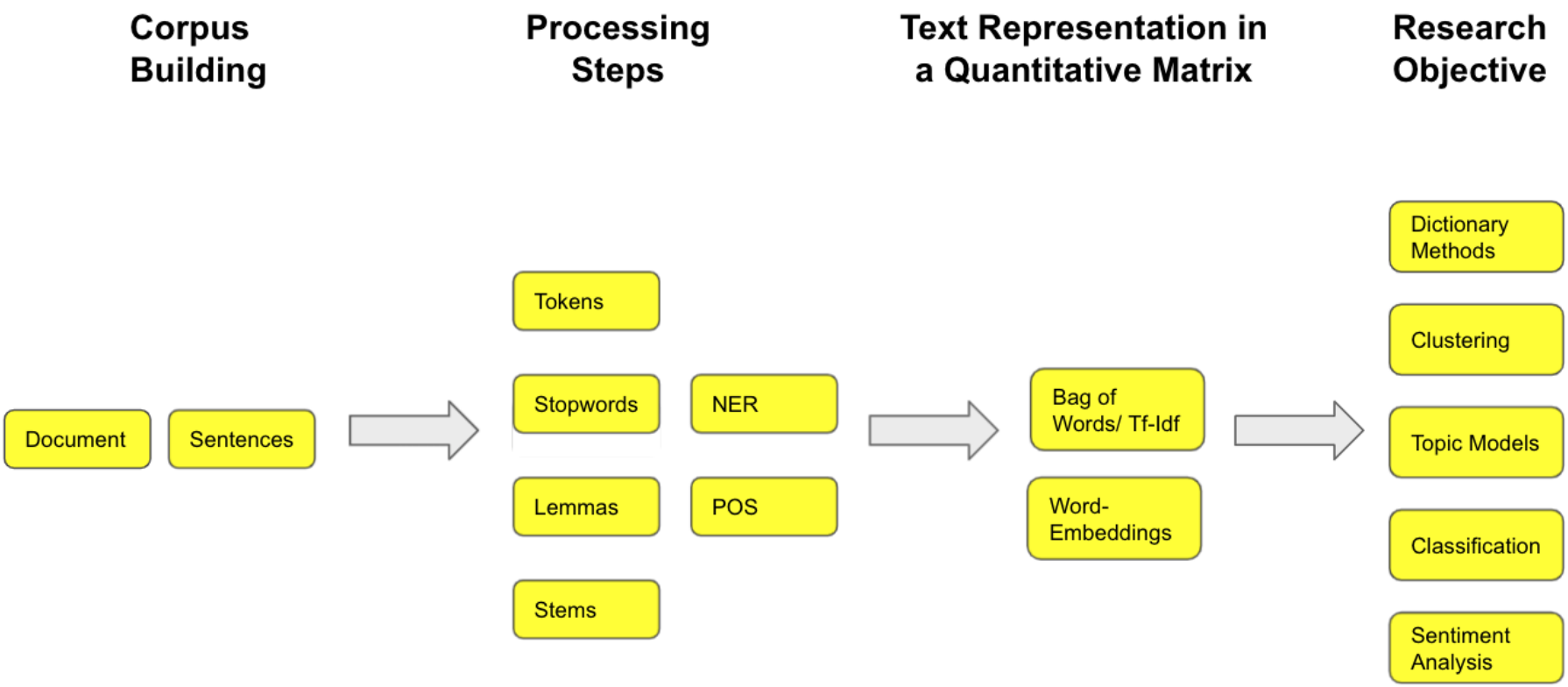
Week 4:
Supervised Machine Learning Technqiues

Ashrakat Elshehawy

Department of Politics and International Relations,
University of Oxford

May 17, 2021

Overview from Text to Data[1]



[1] Slides for the session are based on Federico Nanni's course of Computational Text Analysis at U Mannheim

Recap - What have we learned until now?

- Text Pre-Processing

Recap - What have we learned until now?

- Text Pre-Processing
- Stop-words

Recap - What have we learned until now?

- Text Pre-Processing
- Stop-words
- Naive vs NLTK Tokenizer

Recap - What have we learned until now?

- Text Pre-Processing
- Stop-words
- Naive vs NLTK Tokenizer
- Lemmatization and Stemming

Recap - What have we learned until now?

- Text Pre-Processing
- Stop-words
- Naive vs NLTK Tokenizer
- Lemmatization and Stemming
- Pos Tagging and NERF

Recap - What have we learned until now?

- Text Pre-Processing
- Stop-words
- Naive vs NLTK Tokenizer
- Lemmatization and Stemming
- Pos Tagging and NERF
- Dictionaries and Word-counting

Recap - What have we learned until now?

- Text Pre-Processing
- Stop-words
- Naive vs NLTK Tokenizer
- Lemmatization and Stemming
- Pos Tagging and NERF
- Dictionaries and Word-counting
- Bag of Words/Tf-Idf

Recap - What have we learned until now?

- Text Pre-Processing
- Stop-words
- Naive vs NLTK Tokenizer
- Lemmatization and Stemming
- Pos Tagging and NERF
- Dictionaries and Word-counting
- Bag of Words/Tf-Idf
- Word Embeddings

Recap - What have we learned until now?

- Text Pre-Processing
- Stop-words
- Naive vs NLTK Tokenizer
- Lemmatization and Stemming
- Pos Tagging and NERF
- Dictionaries and Word-counting
- Bag of Words/Tf-Idf
- Word Embeddings
- Cosine-Similarity

Recap - What have we learned until now?

- Text Pre-Processing
- Stop-words
- Naive vs NLTK Tokenizer
- Lemmatization and Stemming
- Pos Tagging and NERF
- Dictionaries and Word-counting
- Bag of Words/Tf-Idf
- Word Embeddings
- Cosine-Similarity
- Unsupervised Techniques

Recap - What have we learned until now?

- Text Pre-Processing
- Stop-words
- Naive vs NLTK Tokenizer
- Lemmatization and Stemming
- Pos Tagging and NERF
- Dictionaries and Word-counting
- Bag of Words/Tf-Idf
- Word Embeddings
- Cosine-Similarity
- Unsupervised Techniques
  - Clustering

Recap - What have we learned until now?

- Text Pre-Processing
- Stop-words
- Naive vs NLTK Tokenizer
- Lemmatization and Stemming
- Pos Tagging and NERF
- Dictionaries and Word-counting
- Bag of Words/Tf-Idf
- Word Embeddings
- Cosine-Similarity
- Unsupervised Techniques
  - Clustering
  - Topic-Modelling

Supervised Machine Learning:

Classification as a Supervised Technique

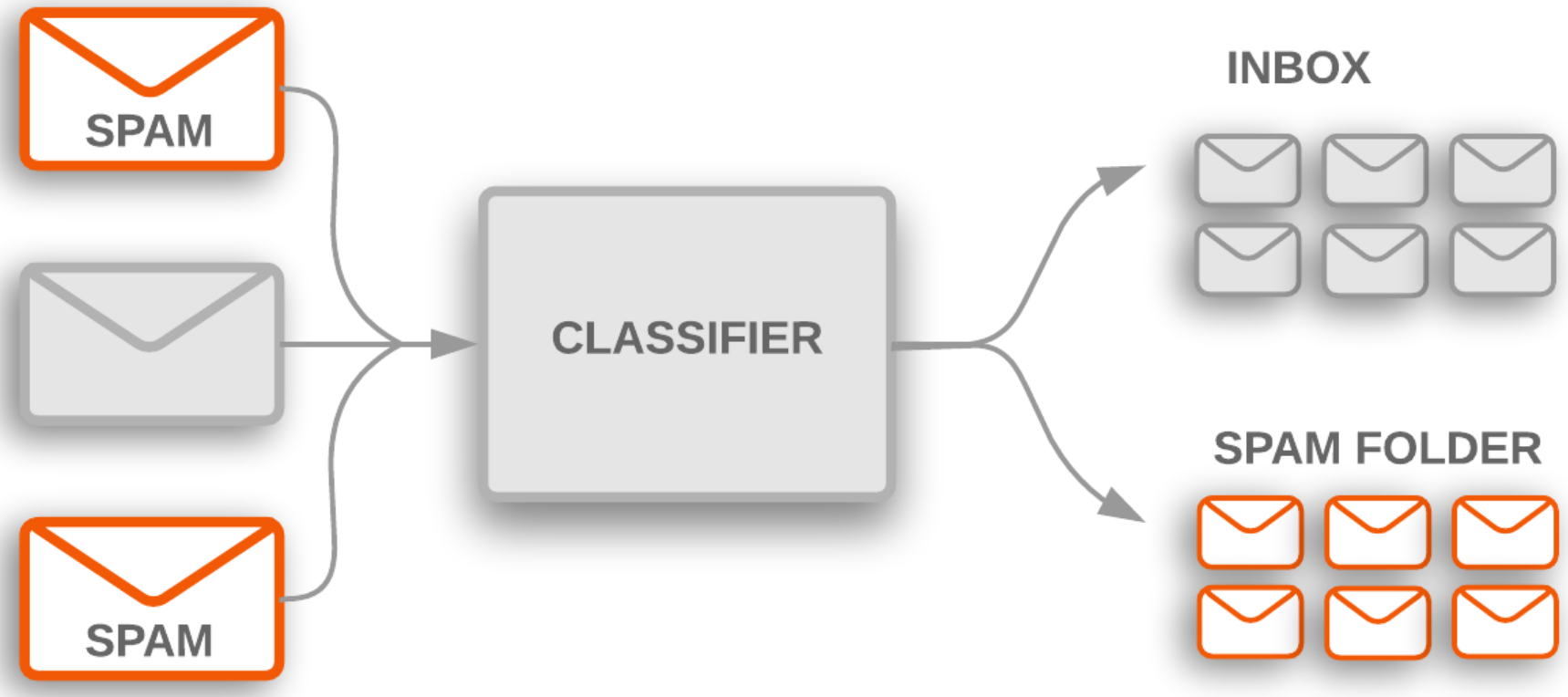


Image source: Source: Google Developers

Classification as a Supervised Technique

What is Classification?

The task of associating instances (e.g., documents, sentences, words) with pre-defined labels

Classification as a Supervised Technique

What is Classification?

The task of associating instances (e.g., documents, sentences, words) with pre-defined labels

Document Classification

Classification as a Supervised Technique

What is Classification?

The task of associating instances (e.g., documents, sentences, words) with pre-defined labels

Document Classification

- Relevance Classification (binary)

Classification as a Supervised Technique

What is Classification?

The task of associating instances (e.g., documents, sentences, words) with pre-defined labels

Document Classification

- Relevance Classification (binary)
- Sentiment analysis (binary / multi-class)

Classification as a Supervised Technique

What is Classification?

The task of associating instances (e.g., documents, sentences, words) with pre-defined labels

Document Classification

- Relevance Classification (binary)
- Sentiment analysis (binary / multi-class)
- Topic Classification (multi-class)

Classification for Sentiment Analysis

This is a text classification task. It is often presented as a binary decision:

- for / against

Classification for Sentiment Analysis

This is a text classification task. It is often presented as a binary decision:

- for / against
- like / dislike

Classification for Sentiment Analysis

This is a text classification task. It is often presented as a binary decision:

- for / against
- like / dislike
- good / bad

Classification: Rule-Based vs. Machine Learning

Rule-based: define a list of "if - then" rules. For example:

Classification: Rule-Based vs. Machine Learning

Rule-based: define a list of "if - then" rules. For example:

- if it mentions "bad" then negative

Classification: Rule-Based vs. Machine Learning

Rule-based: define a list of "if - then" rules. For example:

- if it mentions "bad" then negative

Machine Learning: manually label the data and train a machine learning model on it.

Classification: Rule-Based vs. Machine Learning

Rule-based: define a list of "if - then" rules. For example:

- if it mentions "bad" then negative

Machine Learning: manually label the data and train a machine learning model on it.

Dictionaries could be used in both settings.

Classification: Rule-Based vs. Machine Learning

Problems with Rule-Based

- You need many rules to cover all cases

Classification: Rule-Based vs. Machine Learning

Problems with Rule-Based

- You need many rules to cover all cases
- You need to handle many exceptions

Classification: Rule-Based vs. Machine Learning

Problems with Rule-Based

- You need many rules to cover all cases
- You need to handle many exceptions
- Defining rules is time consuming and inherently subjective

Classification: Rule-Based vs. Machine Learning

Problems with Dictionaries

- Time consuming to create

Classification: Rule-Based vs. Machine Learning

Problems with Dictionaries

- Time consuming to create
- Difficult to have complete coverage of the vocabulary used

Classification: Rule-Based vs. Machine Learning

Problems with Dictionaries

- Time consuming to create
- Difficult to have complete coverage of the vocabulary used
- Often topic-dependent

Classification: Rule-Based vs. Machine Learning

Problems with Machine Learning

- Labeling is extremely time consuming and tedious

Classification: Rule-Based vs. Machine Learning

Problems with Machine Learning

- Labeling is extremely time consuming and tedious
- Trained models are difficult to interpret

Classification: Rule-Based vs. Machine Learning

Problems with Machine Learning

- Labeling is extremely time consuming and tedious
- Trained models are difficult to interpret
- Some algorithms work better on certain tasks (no free lunch theorem)

Classification: Labelling

- Define the specific task (e.g., "Identifying the polarity of Donald Trump's tweets")

Classification: Labelling

- Define the specific task (e.g., "Identifying the polarity of Donald Trump's tweets")
- Prepare guidelines for labeling consistently

Classification: Labelling

- Define the specific task (e.g., "Identifying the polarity of Donald Trump's tweets")
- Prepare guidelines for labeling consistently
- Having different (at least 3) annotators

Classification: Labelling

- Define the specific task (e.g., "Identifying the polarity of Donald Trump's tweets")
- Prepare guidelines for labeling consistently
- Having different (at least 3) annotators
- Compute their agreement (to know how difficult it is)

Classification: Labelling

- Define the specific task (e.g., "Identifying the polarity of Donald Trump's tweets")
- Prepare guidelines for labeling consistently
- Having different (at least 3) annotators
- Compute their agreement (to know how difficult it is)

$\rightarrow$ Bad labels generate the *"Garbage-in-Garbage-out"* effect!

Classification: Labelling

- Define the specific task (e.g., "Identifying the polarity of Donald Trump's tweets")
- Prepare guidelines for labeling consistently
- Having different (at least 3) annotators
- Compute their agreement (to know how difficult it is)

→ Bad labels generate the *"Garbage-in-Garbage-out"* effect!

→ A (properly) labelled dataset is called gold standard (or ground-truth

Classification: Evaluation

Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Classification: Evaluation

Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Measures:

Classification: Evaluation

Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Measures:

- Accuracy: proportion of correct results over total

Classification: Evaluation

Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Measures:

- Accuracy: proportion of correct results over total
- Precision: number of selected items that are relevant

Classification: Evaluation

Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Measures:

- Accuracy: proportion of correct results over total
- Precision: number of selected items that are relevant
- Recall: number of relevant items that are selected

Classification: Evaluation

Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Measures:

- Accuracy: proportion of correct results over total
- Precision: number of selected items that are relevant
- Recall: number of relevant items that are selected
- F-1 Score: the harmonic mean of precision and recall

Classification: Evaluation Example

Gold Standard: 200 labelled tweets from Trump, 170 negative, 30 positive

Classification: Evaluation Example

Gold Standard: 200 labelled tweets from Trump, 170 negative, 30 positive

System output: 195 tweets negative (169 corrects), 5 positive (4 corrects)

Classification: Evaluation Example

Gold Standard: 200 labelled tweets from Trump, 170 negative, 30 positive

System output: 195 tweets negative (169 corrects), 5 positive (4 corrects)

- Accuracy: proportion of correct results over total

Classification: Evaluation Example

Gold Standard: 200 labelled tweets from Trump, 170 negative, 30 positive

System output: 195 tweets negative (169 corrects), 5 positive (4 corrects)

- Accuracy: proportion of correct results over total
  - Negative: 99%, Positive: 13%

Classification: Evaluation Example

Gold Standard: 200 labelled tweets from Trump, 170 negative, 30 positive

System output: 195 tweets negative (169 corrects), 5 positive (4 corrects)

- Accuracy: proportion of correct results over total
  - Negative: 99%, Positive: 13%
- Precision: number of selected items that are relevant

Classification: Evaluation Example

Gold Standard: 200 labelled tweets from Trump, 170 negative, 30 positive

System output: 195 tweets negative (169 corrects), 5 positive (4 corrects)

- Accuracy: proportion of correct results over total
  - Negative: 99%, Positive: 13%
- Precision: number of selected items that are relevant
  - Negative: 87%, Positive: 80%

Classification: Evaluation Example

Gold Standard: 200 labelled tweets from Trump, 170 negative, 30 positive

System output: 195 tweets negative (169 corrects), 5 positive (4 corrects)

- Accuracy: proportion of correct results over total
  - Negative: 99%, Positive: 13%
- Precision: number of selected items that are relevant
  - Negative: 87%, Positive: 80%
- Recall: number of relevant items that are selected

Classification: Evaluation Example

Gold Standard: 200 labelled tweets from Trump, 170 negative, 30 positive

System output: 195 tweets negative (169 corrects), 5 positive (4 corrects)

- Accuracy: proportion of correct results over total
  - Negative: 99%, Positive: 13%
- Precision: number of selected items that are relevant
  - Negative: 87%, Positive: 80%
- Recall: number of relevant items that are selected
  - Negative: 99%, Positive: 13%

Classification: Evaluation Example

Gold Standard: 200 labelled tweets from Trump, 170 negative, 30 positive

System output: 195 tweets negative (169 corrects), 5 positive (4 corrects)

- Accuracy: proportion of correct results over total
  - Negative: 99%, Positive: 13%
- Precision: number of selected items that are relevant
  - Negative: 87%, Positive: 80%
- Recall: number of relevant items that are selected
  - Negative: 99%, Positive: 13%
- F-1 Score: the harmonic mean of precision and recall

Classification: Evaluation Example

Gold Standard: 200 labelled tweets from Trump, 170 negative, 30 positive

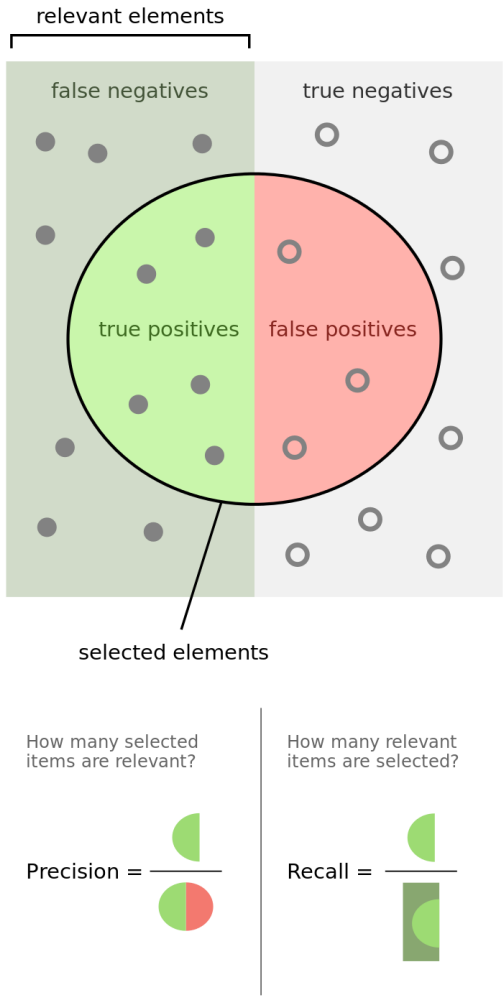
System output: 195 tweets negative (169 corrects), 5 positive (4 corrects)

- Accuracy: proportion of correct results over total
  - Negative: 99%, Positive: 13%
- Precision: number of selected items that are relevant
  - Negative: 87%, Positive: 80%
- Recall: number of relevant items that are selected
  - Negative: 99%, Positive: 13%
- F-1 Score: the harmonic mean of precision and recall
  - Negative: 93%, Positive: 22%

Classification: Precision and Recall

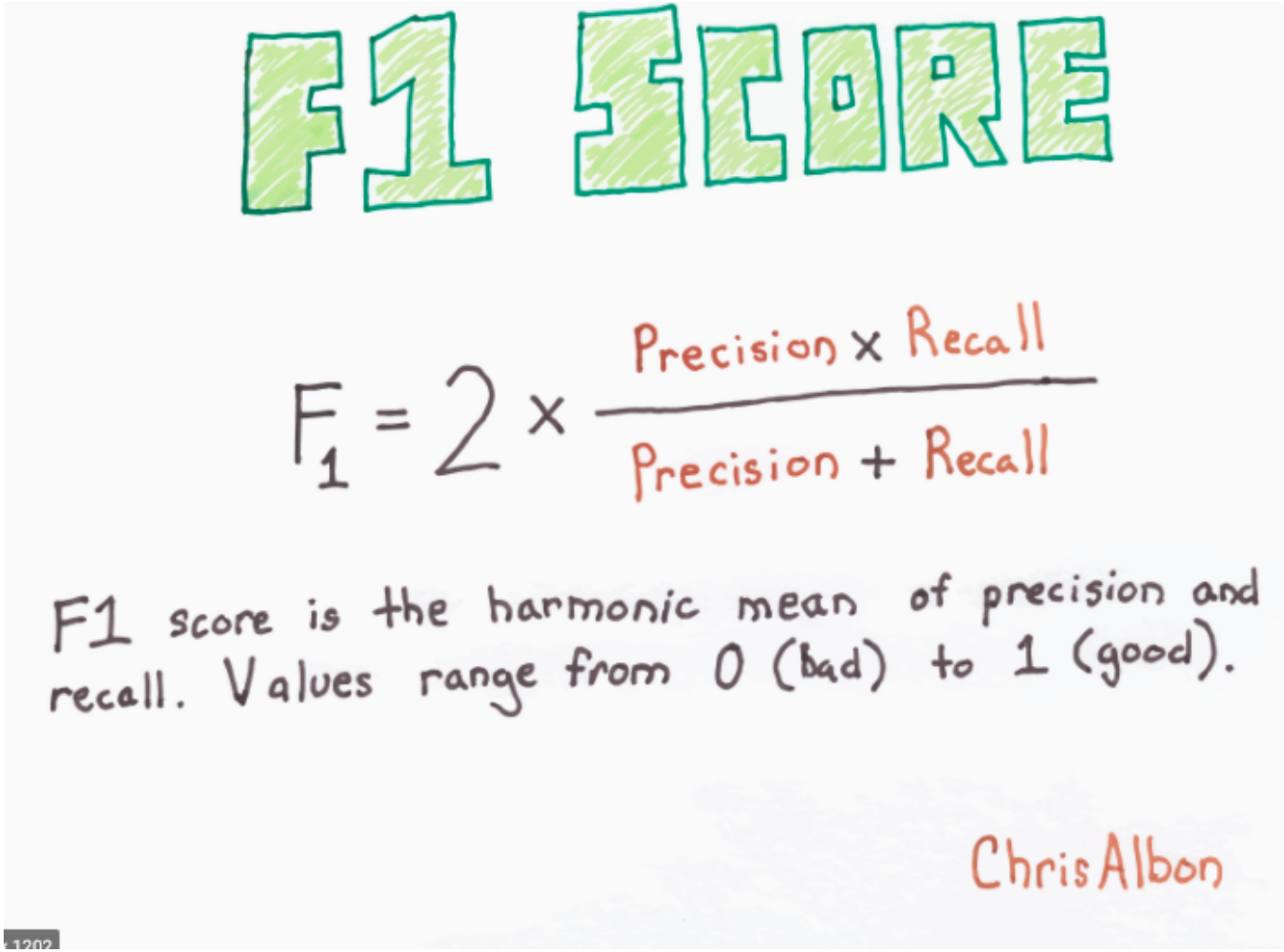
Precision: (true positives / all positives)

Recall: (true positives / relevant elements)



Source: Wikipedia

Classification: F1 score



How does supervised machine learning work?

- You provide the system (lots of) examples of the things you want to classify (we call it training set), in pair: instance - label

How does supervised machine learning work?

- You provide the system (lots of) examples of the things you want to classify (we call it training set), in pair: instance - label
- You train the system (.fit()): it will try to learn how to classify instances in labels in order to minimize a loss function

How does supervised machine learning work?

- You provide the system (lots of) examples of the things you want to classify (we call it training set), in pair: instance - label
- You train the system (.fit()): it will try to learn how to classify instances in labels in order to minimize a loss function
- You test on a different dataset (test set), to measure performances

Over-fitting and Under-fitting[2]

- Underfit Model is simple because of the assumptions made about the data pays very little attention to the training data and oversimplifies the model

[2] Source: IT Bodhi Medium

Over-fitting and Under-fitting[2]

- Underfit Model is simple because of the assumptions made about the data pays very little attention to the training data and oversimplifies the model

- Overfitting means your model is not Generalised. Algorithm used to build prediction model is very complex - it has over learned underlying patterns in training data

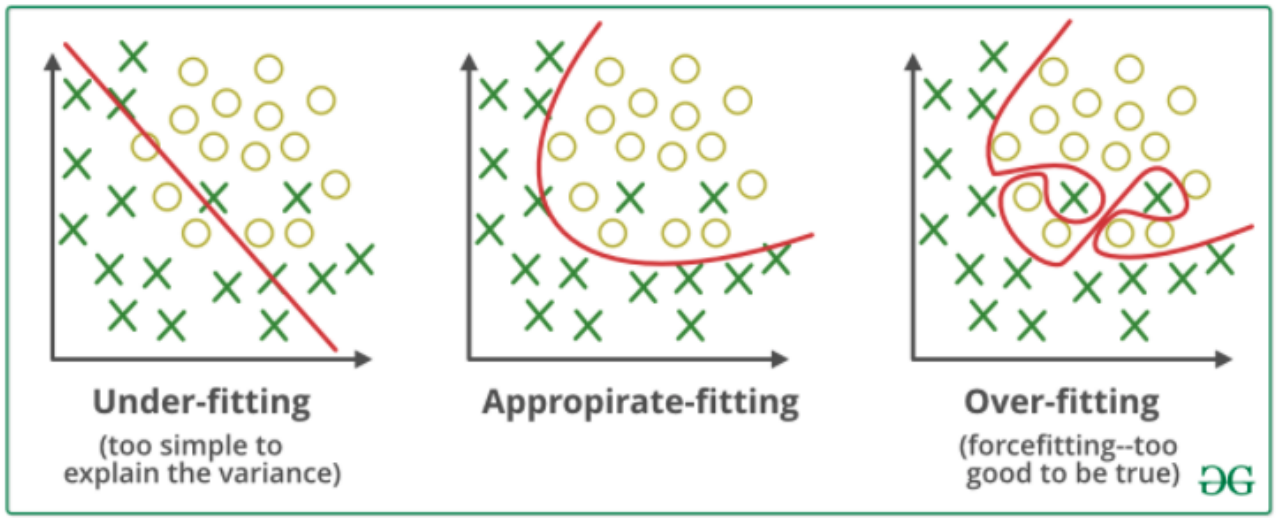


Figure: `https://www.geeksforgeeks.org/underfitting-and-overfitting-in-machine-learning/`

[2] Source: IT Bodhi Medium

Figure: Source:AnalyticsVidhya

Training and Test Sets

1) You split your gold standard in training and test (and validation)

Training and Test Sets

1) You split your gold standard in training and test (and validation)

Training and Test Sets

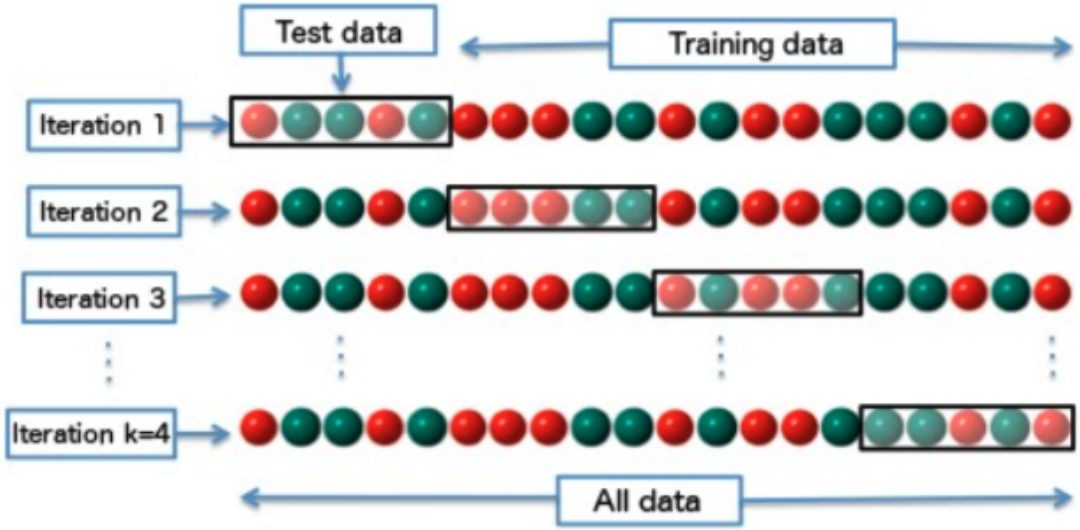
1) You split your gold standard in training and test (and validation)

2)You use cross-validation

Training and Test Sets

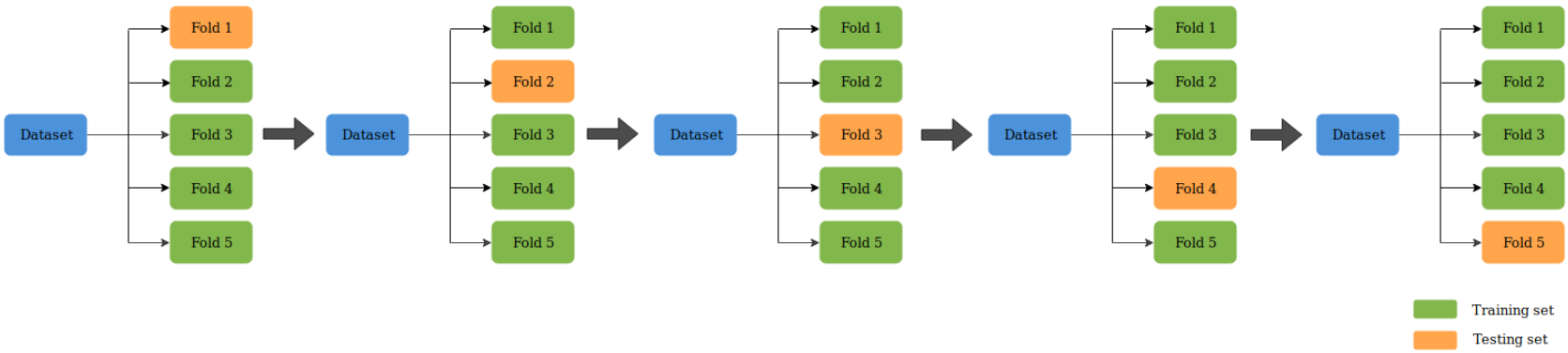
1) You split your gold standard in training and test (and validation)

2)You use cross-validation



K-Folds Cross Validation Summary[3]

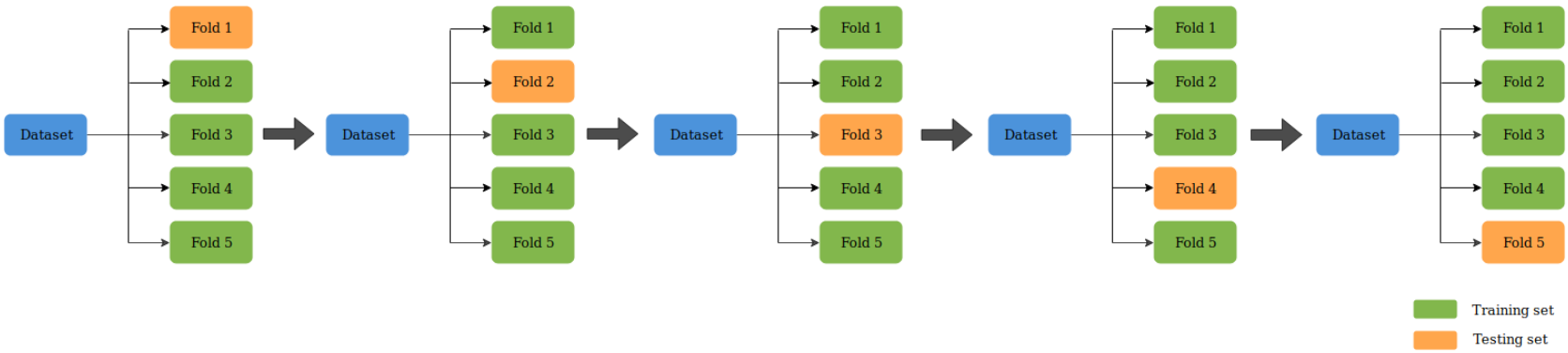
1. Shuffle your data randomly



[3] Read more here: Link to Tutorial on SVM by Hackerearth

K-Folds Cross Validation Summary[3]

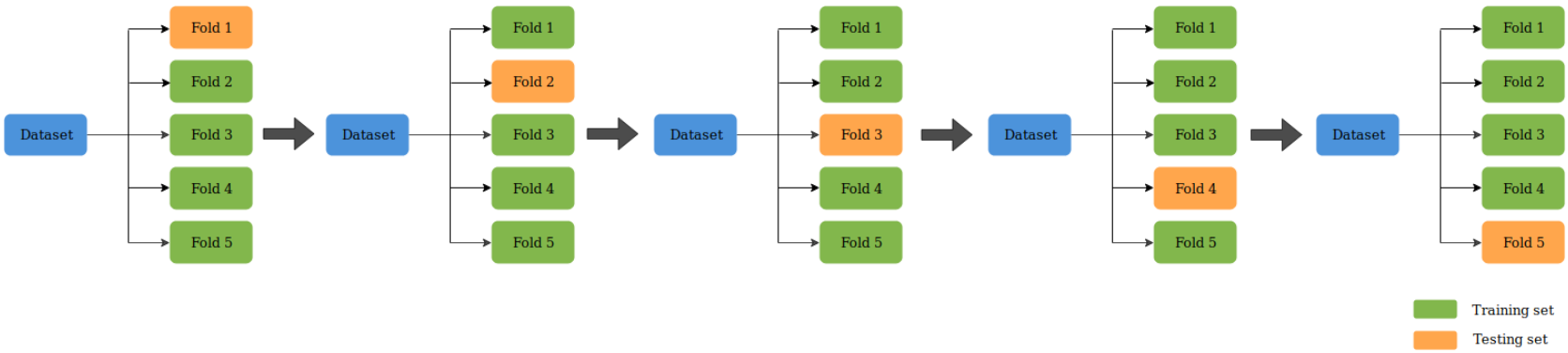
1. Shuffle your data randomly
2. Split your entire data into k groups



[3] Read more here: Link to Tutorial on SVM by Hackerearth

K-Folds Cross Validation Summary[3]

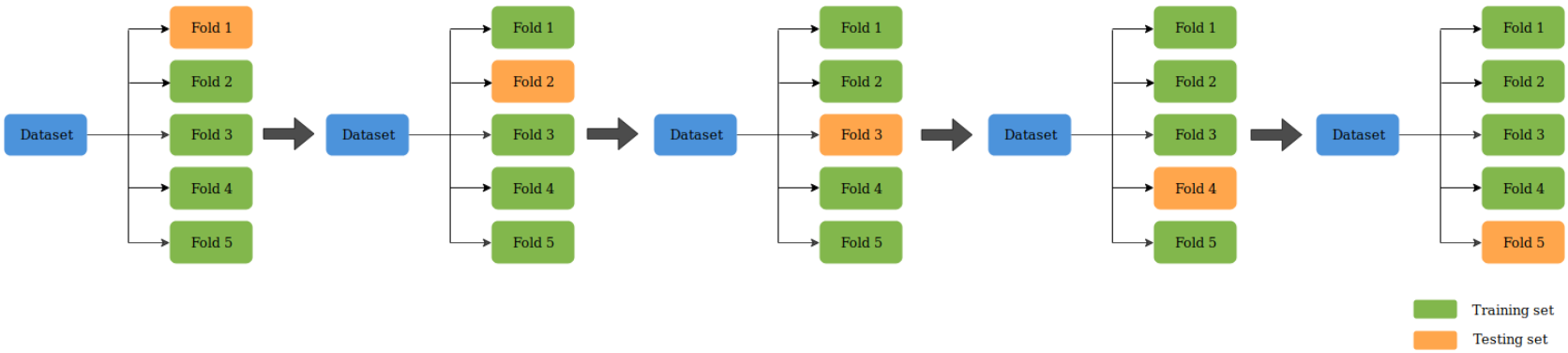
1. Shuffle your data randomly
2. Split your entire data into k groups
3. For each unique group:



[3] Read more here: Link to Tutorial on SVM by Hackerearth

K-Folds Cross Validation Summary[3]

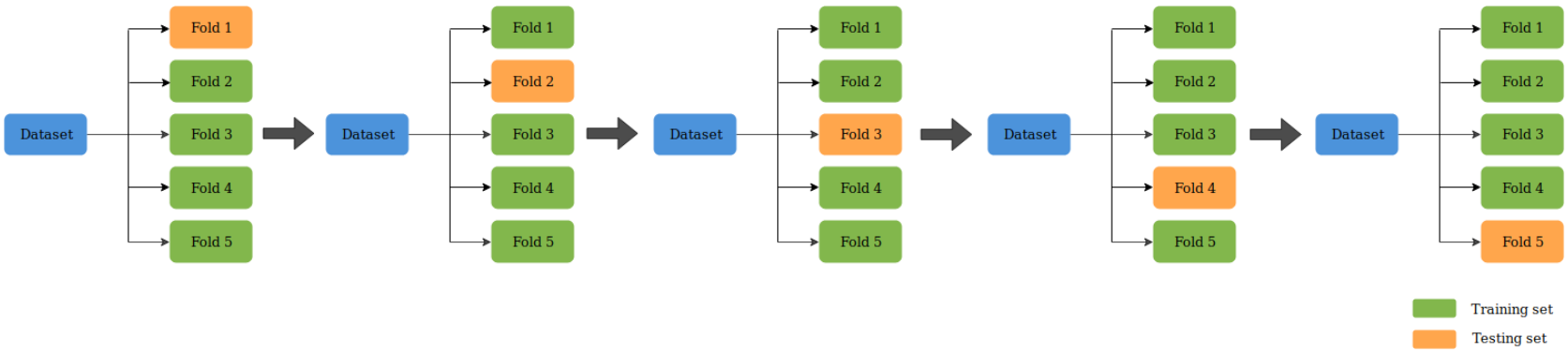
1. Shuffle your data randomly
2. Split your entire data into k groups
3. For each unique group:
   1. Take the group as a hold out or test data set



[3] Read more here: Link to Tutorial on SVM by Hackerearth

K-Folds Cross Validation Summary[3]

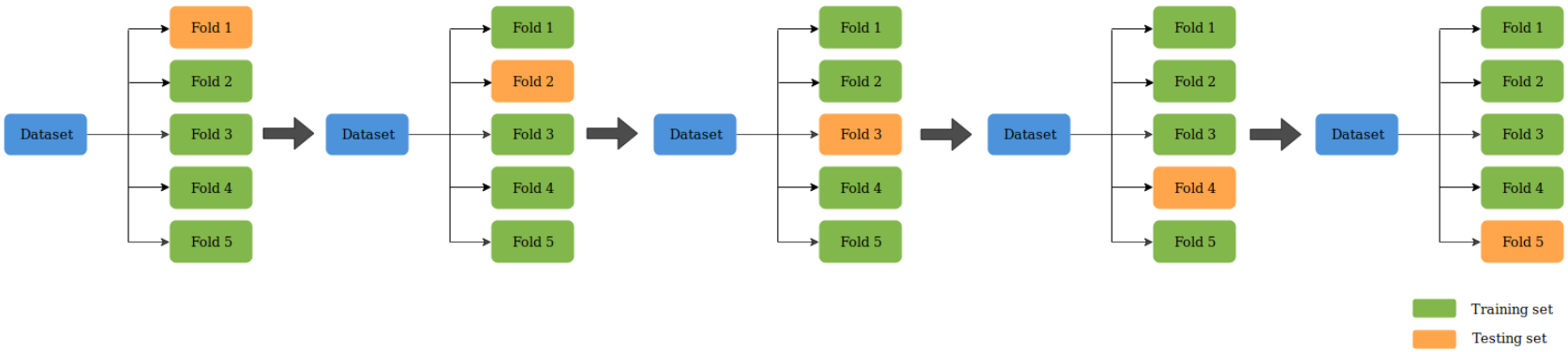
1. Shuffle your data randomly
2. Split your entire data into k groups
3. For each unique group:
   1. Take the group as a hold out or test data set
   2. Take the remaining groups as a training data set



[3] Read more here: Link to Tutorial on SVM by Hackerearth

K-Folds Cross Validation Summary[3]

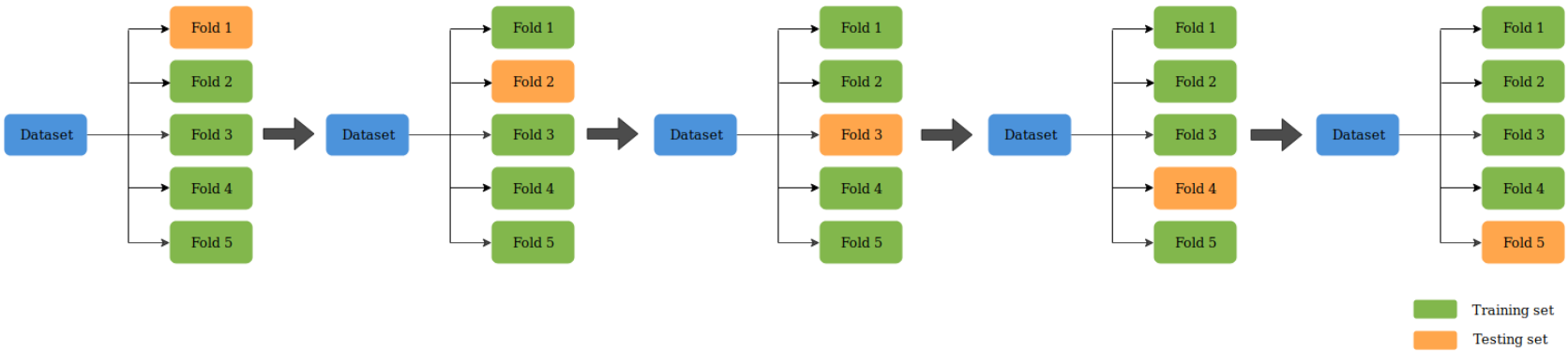
1. Shuffle your data randomly
2. Split your entire data into k groups
3. For each unique group:
   1. Take the group as a hold out or test data set
   2. Take the remaining groups as a training data set
   3. Fit a model on the training set and evaluate it on the test set



[3] Read more here: Link to Tutorial on SVM by Hackerearth

K-Folds Cross Validation Summary[3]

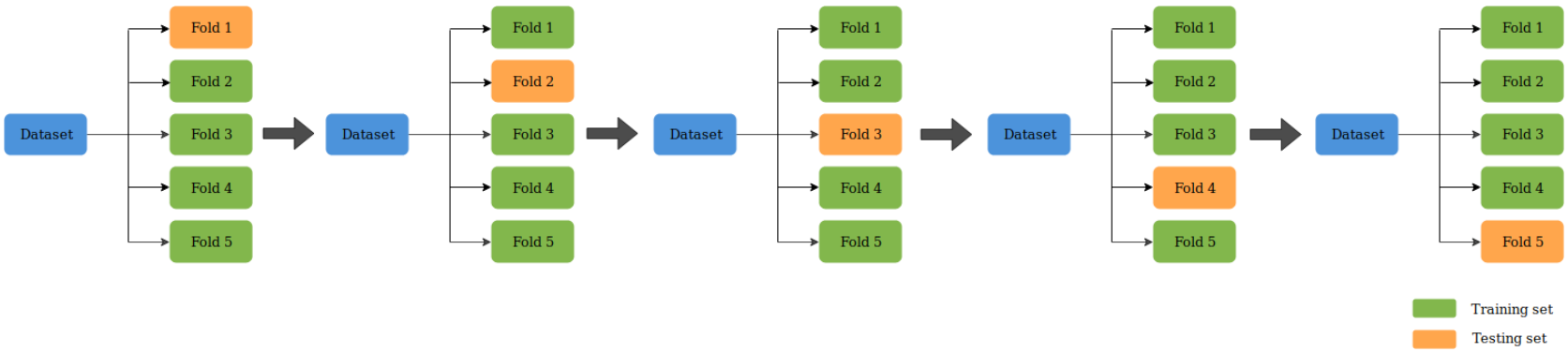
1. Shuffle your data randomly
2. Split your entire data into k groups
3. For each unique group:
   1. Take the group as a hold out or test data set
   2. Take the remaining groups as a training data set
   3. Fit a model on the training set and evaluate it on the test set
   4. Retain the evaluation score and discard the model



[3] Read more here: Link to Tutorial on SVM by Hackerearth

K-Folds Cross Validation Summary[3]

1. Shuffle your data randomly
2. Split your entire data into k groups
3. For each unique group:
   1. Take the group as a hold out or test data set
   2. Take the remaining groups as a training data set
   3. Fit a model on the training set and evaluate it on the test set
   4. Retain the evaluation score and discard the model
4. Evaluate the Model using the measures we have learned (Accuracy, Precision, Recall, F1 score)



[3] Read more here: Link to Tutorial on SVM by Hackerearth

Evaluate

Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Evaluate

Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Measures:

Evaluate

Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Measures:

- Accuracy: proportion of correct results over total

Evaluate

Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Measures:

- Accuracy: proportion of correct results over total
- Precision: number of selected items that are relevant

Evaluate

Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Measures:

- Accuracy: proportion of correct results over total
- Precision: number of selected items that are relevant
- Recall: number of relevant items that are selected

Evaluate

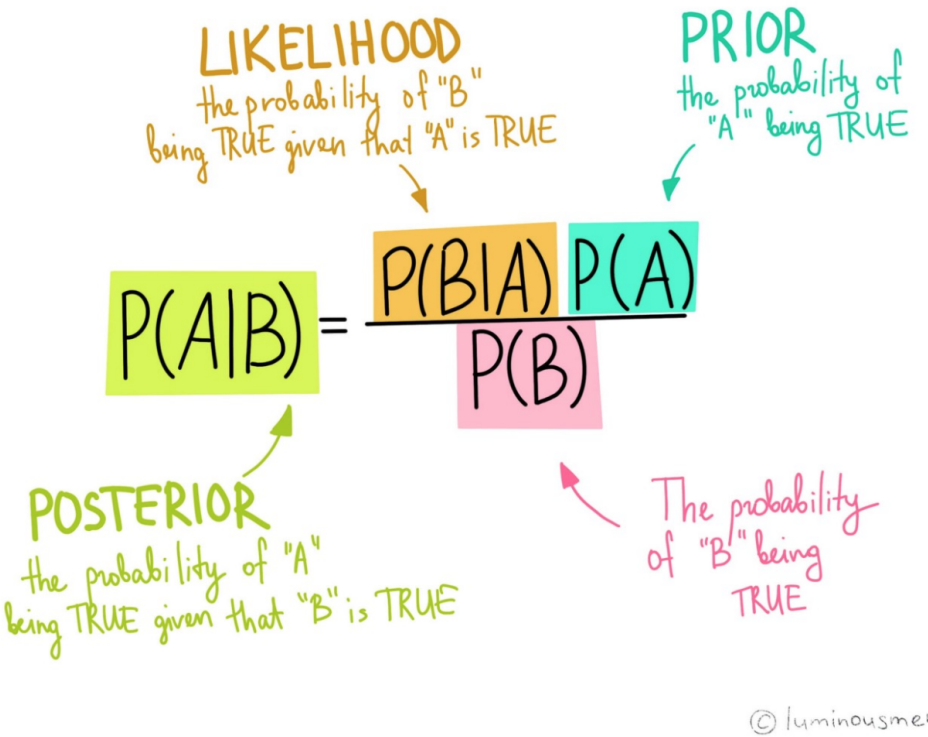
Essential to know if your rule-based, dictionary-based or machine learning system works properly.

Measures:

- Accuracy: proportion of correct results over total
- Precision: number of selected items that are relevant
- Recall: number of relevant items that are selected
- F-1 Score: the harmonic mean of precision and recall

Naive Bayes Classifier

A probabilistic generative model: outputs are class/label probabilities.



$$P(\text{label} \mid \text{word}_1 \ldots \text{word}_n) = \frac{P(\text{word}_1 \ldots \text{word}_n \mid \text{label}) * P(\text{label})}{P(\text{word}_1 \ldots \text{word}_n)}$$

What you want to know

What you see

Computed on the training data

Compared to other labels

Computed on the training data

Naive Bayes Pros and Cons

Pros:

Naive Bayes Pros and Cons

Pros:

- Simple to train

Naive Bayes Pros and Cons

Pros:

- Simple to train
- Provides probability outputs (for each feature!)

Naive Bayes Pros and Cons

Pros:

- Simple to train
- Provides probability outputs (for each feature!)
- Works on multi-class by default

Naive Bayes Pros and Cons

Pros:

- Simple to train
- Provides probability outputs (for each feature!)
- Works on multi-class by default

Cons:

Naive Bayes Pros and Cons

Pros:

- Simple to train
- Provides probability outputs (for each feature!)
- Works on multi-class by default

Cons:

- No feature-dependence ("new" and "york" are independent tokens)

Naive Bayes Pros and Cons

Pros:

- Simple to train
- Provides probability outputs (for each feature!)
- Works on multi-class by default

Cons:

- No feature-dependence ("new" and "york" are independent tokens)
- Not powerful enough to model complex feature interactions

Naive Bayes Pros and Cons

Pros:

- Simple to train
- Provides probability outputs (for each feature!)
- Works on multi-class by default

Cons:

- No feature-dependence ("new" and "york" are independent tokens)
- Not powerful enough to model complex feature interactions

Take away: Often used as a competitive baseline for text classification tasks

Support Vector Machine (SVM) Classifier

Non-probabilistic discriminative model.

Support Vector Machine (SVM) Classifier

Non-probabilistic discriminative model.

- Project instances in space.

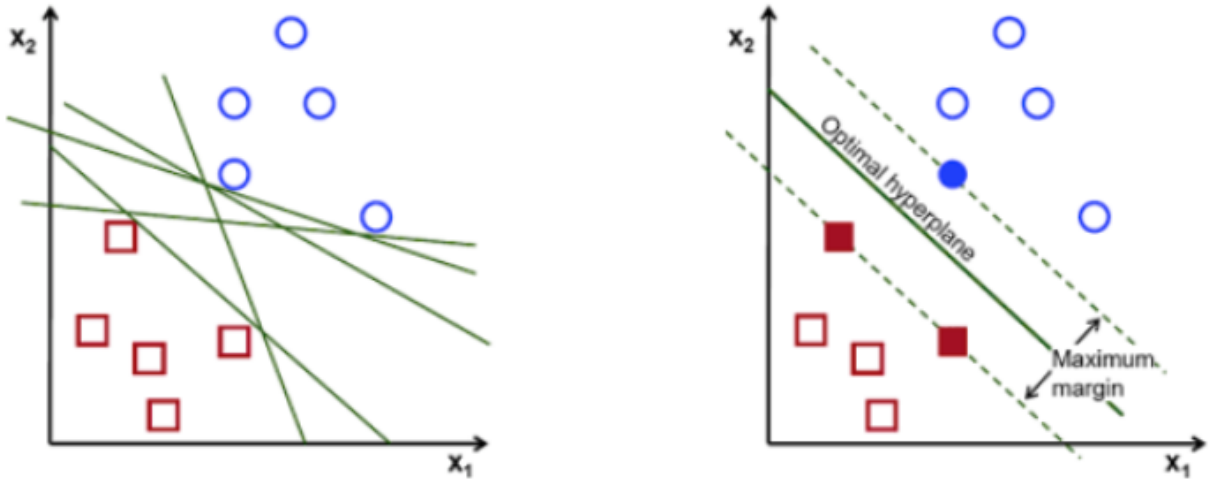


Figure: Source: towardsdatascience.com

Support Vector Machine (SVM) Classifier

Non-probabilistic discriminative model.

- Project instances in space.
- Find hyperplane that maximizes margin between closest opposite examples.

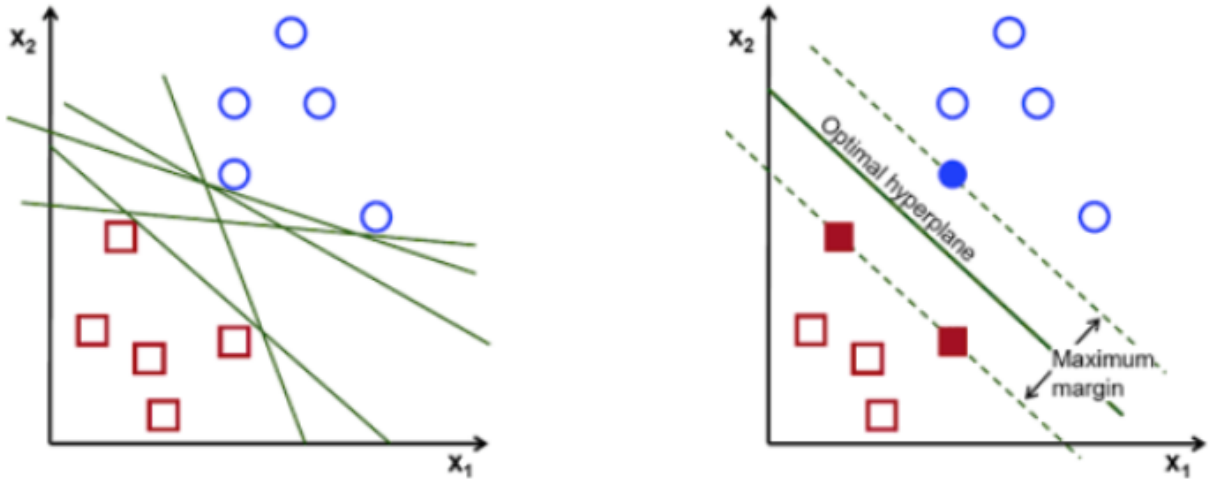


Figure: Source: towardsdatascience.com

Support Vector Machine (SVM) Classifier

Non-probabilistic discriminative model.

- Project instances in space.
- Find hyperplane that maximizes margin between closest opposite examples.

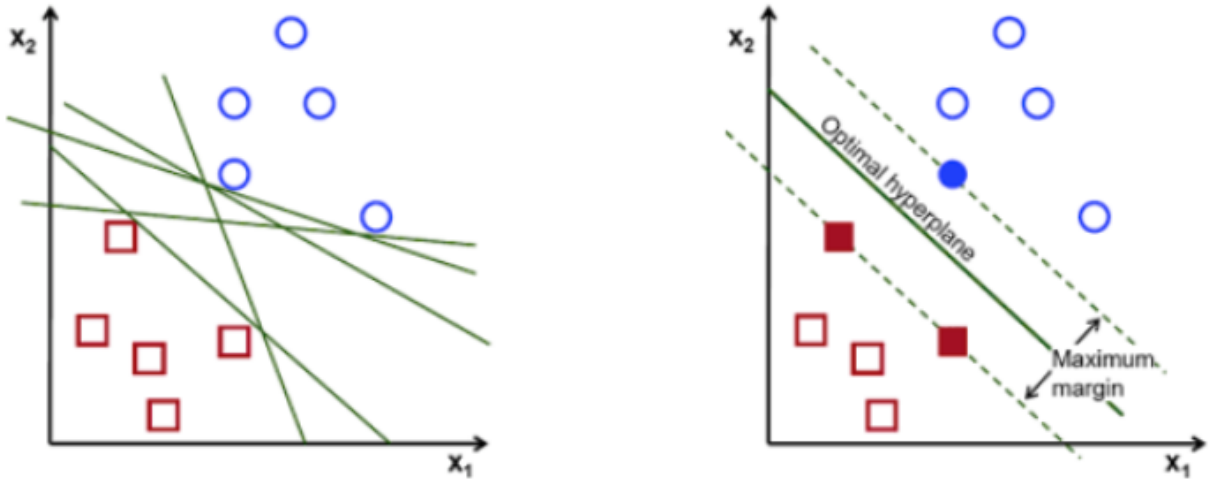


Figure: Source: towardsdatascience.com

- Kernel-trick = if no optimal hyperplane, maps examples to a higher-dim. space

Further on SVM

The parameter C determines how much you want to avoid misclassifying each training example.

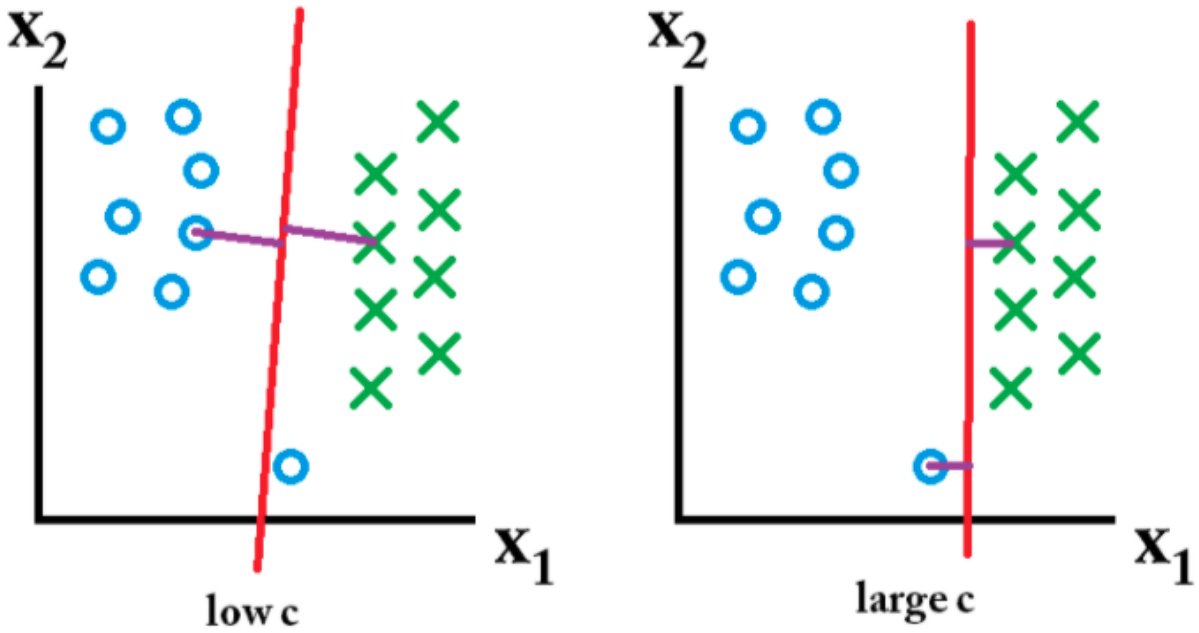


Figure: Source: From:https://stats.stackexchange.com/questions/31066/what-is-the-influence-of-c-in-svms-with-linear-kernel

SVM Pros and Cons

Pros:

SVM Pros and Cons

Pros:

- Generally good at generalizing
- Efficient (you only store the support vectors)

SVM Pros and Cons

Pros:

- Generally good at generalizing
- Efficient (you only store the support vectors)

SVM Pros and Cons

Pros:

- Generally good at generalizing
- Efficient (you only store the support vectors)

Cons:

SVM Pros and Cons

Pros:

- Generally good at generalizing
- Efficient (you only store the support vectors)

Cons:

- Extension to multi-class not obvious (one-vs-one, one-vs-rest)

SVM Pros and Cons

Pros:

- Generally good at generalizing
- Efficient (you only store the support vectors)

Cons:

- Extension to multi-class not obvious (one-vs-one, one-vs-rest)
- Several parameters to tune:

SVM Pros and Cons

Pros:

- Generally good at generalizing
- Efficient (you only store the support vectors)

Cons:

- Extension to multi-class not obvious (one-vs-one, one-vs-rest)
- Several parameters to tune:
  - Regularization parameter C

SVM Pros and Cons

Pros:

- Generally good at generalizing
- Efficient (you only store the support vectors)

Cons:

- Extension to multi-class not obvious (one-vs-one, one-vs-rest)
- Several parameters to tune:
  - Regularization parameter C
  - Kernel used (linear, polynomial, RBF)

SVM Pros and Cons

Pros:

- Generally good at generalizing
- Efficient (you only store the support vectors)

Cons:

- Extension to multi-class not obvious (one-vs-one, one-vs-rest)
- Several parameters to tune:
  - Regularization parameter C
  - Kernel used (linear, polynomial, RBF)
  - Kernel parameters