Lancaster University

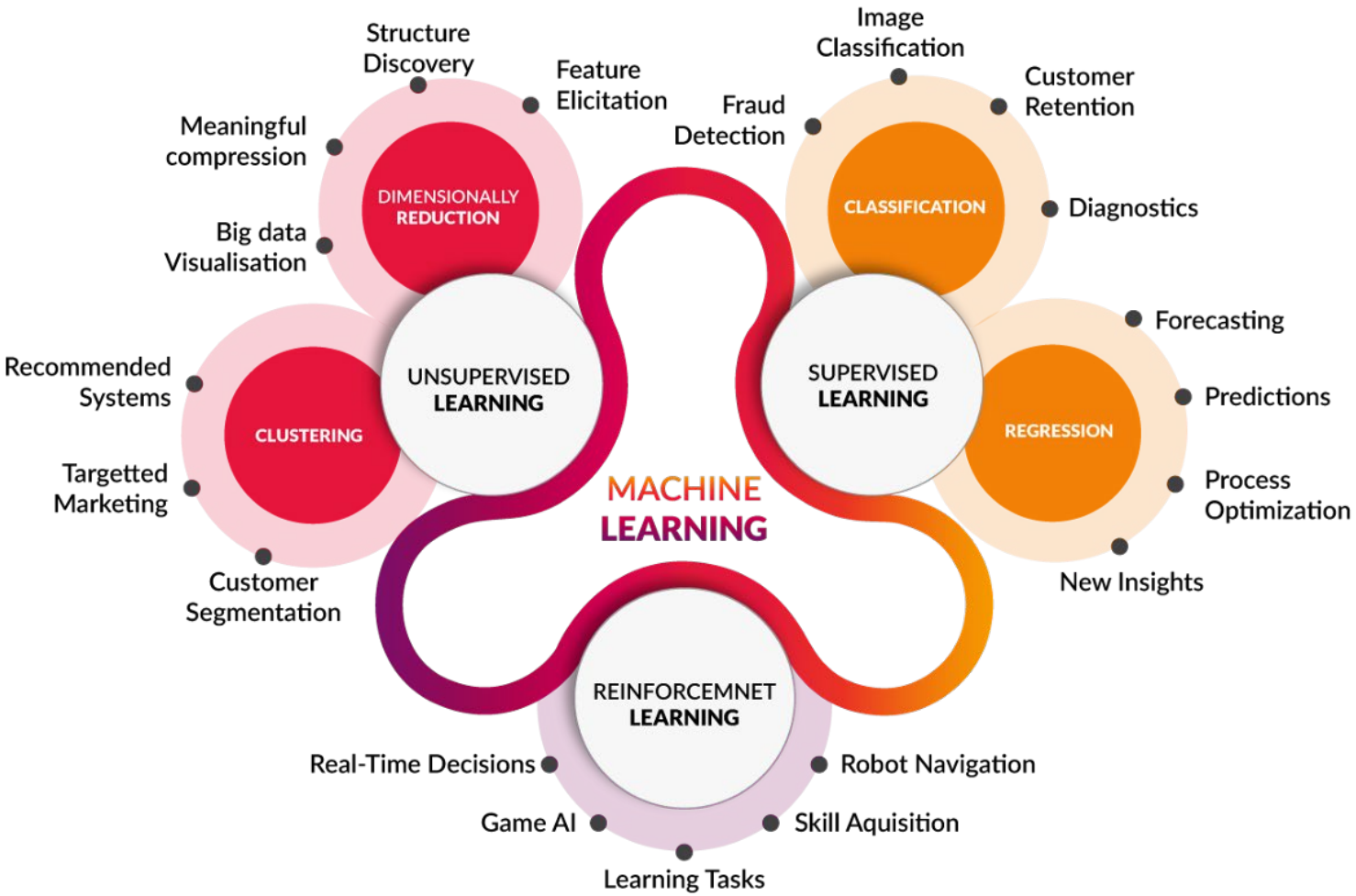
# Introduction to Machine Learning and Deep Learning

**Dr Mahmoud El-Haj**
**SCC, Lancaster University**

InfoLab21

Office C28, InfoLab21
m.el-haj@Lancaster.ac.uk

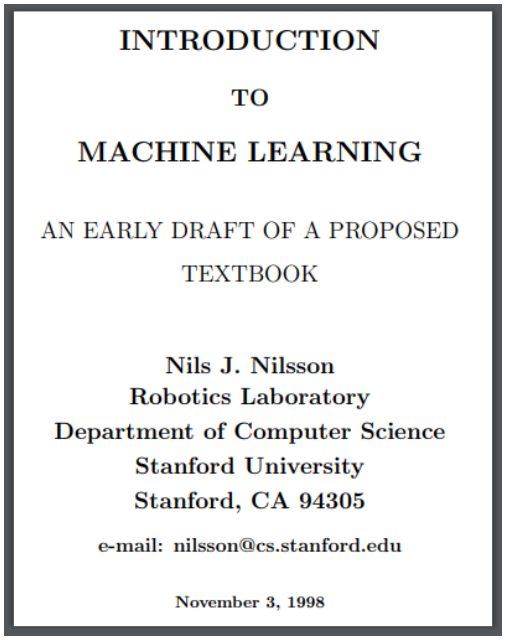
# MACHINE LEARNING



# Relevant Book for Machine Learning

- Introduction to Machine Learning: An Early Draft of a Proposed Textbook. N. J. Nilsson, Stanford.

http://ai.stanford.edu/people/nilsson/mlbook.html



INTRODUCTION

TO

MACHINE LEARNING

AN EARLY DRAFT OF A PROPOSED TEXTBOOK

Nils J. Nilsson
Robotics Laboratory
Department of Computer Science
Stanford University
Stanford, CA 94305

e-mail: nilsson@cs.stanford.edu

November 3, 1998

# Your Learning Outcomes

- Understand fundamental Machine Learning concepts and current issues.

- Understand the difference between learning methods, algorithms and how to apply that on Automatic Text Classification.

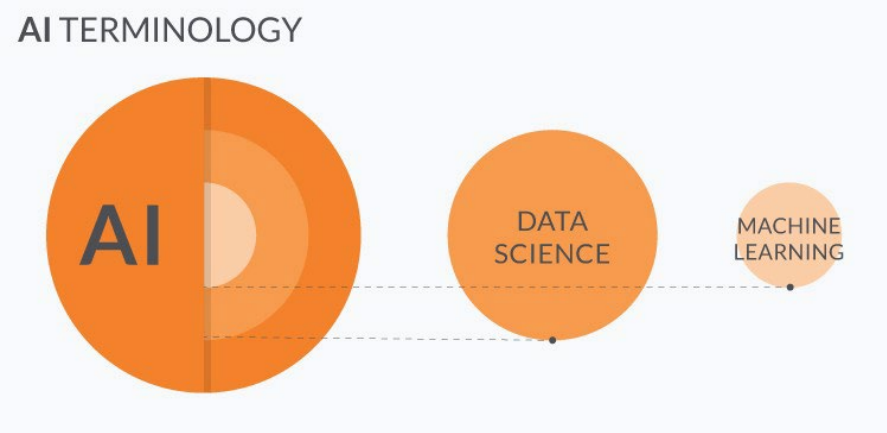
# MACHINE LEARNING – FIRST THINGS FIRST

- Machine:

  Semi or fully automated device that magnifies human physical and/or mental capabilities in performing one or more operations.

- Where does NLP fit into AI?

  Machine Learning is a current application of AI



Lancaster University

# Part 0:
# Introduction to Machine Learning

History
Definitions
Applications
Timeline
Philosophy

# What is Learning?

- Learning:

  "to gain knowledge,
  or understanding of,
  or skill in,
  by study, instruction, or experience,"
  and "modification of a behavioural tendency by experience."

  (*Nils Nilsson*)

- Definition: "improving performance in some task with experience"

  (*Tom Mitchell).*

# What is Machine Learning?

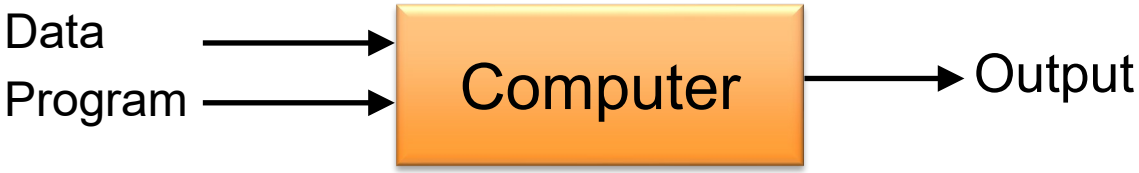
- broadly: a machine learns whenever it changes its structure, program, or data (based on its inputs or in response to external information) in such a manner that its expected future performance improves.

- *Arthur Samuel (1959): "Field of study that gives computers the ability to learn without being explicitly programmed".*

- *He wrote a program that learnt to play checkers (BE: draughts)*

# Machine Learning vs Programming

## Traditional Programming

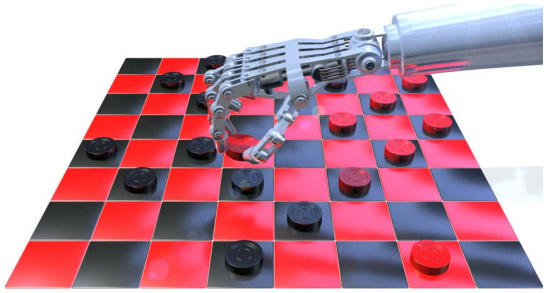



## Machine Learning

# What is Machine Learning?

- *Tom Mitchel (1998): a computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E;*

- *T: Task of playing checkers.*
- *E: Experience of having the program to play 10s of thousands of games against itself.*
- *P: Probability of winning the next game of checkers against a new opponent.*



# What is Machine Learning?

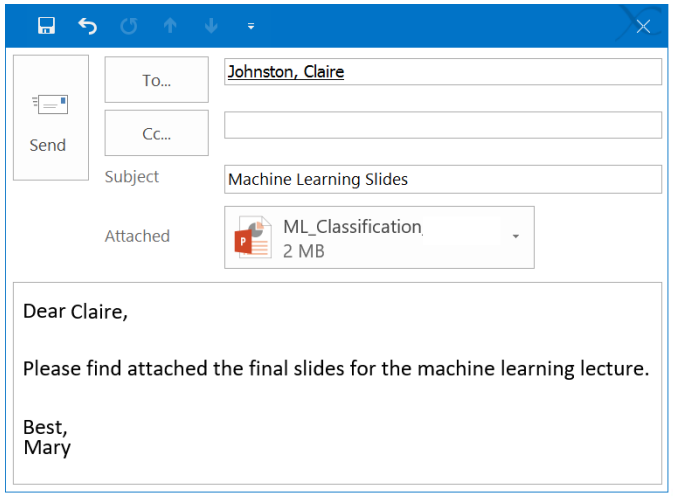
- *Suppose your email application watches which emails you attach or don't attach files to, and based on that it learns how to better remind you if you forgot to attach a file. Can you identify T, E and P in this example?*

T ❑ *Classify whether an email needs or does not need an attachment*

E ❑ *Watching you attaching or not attaching files to emails*

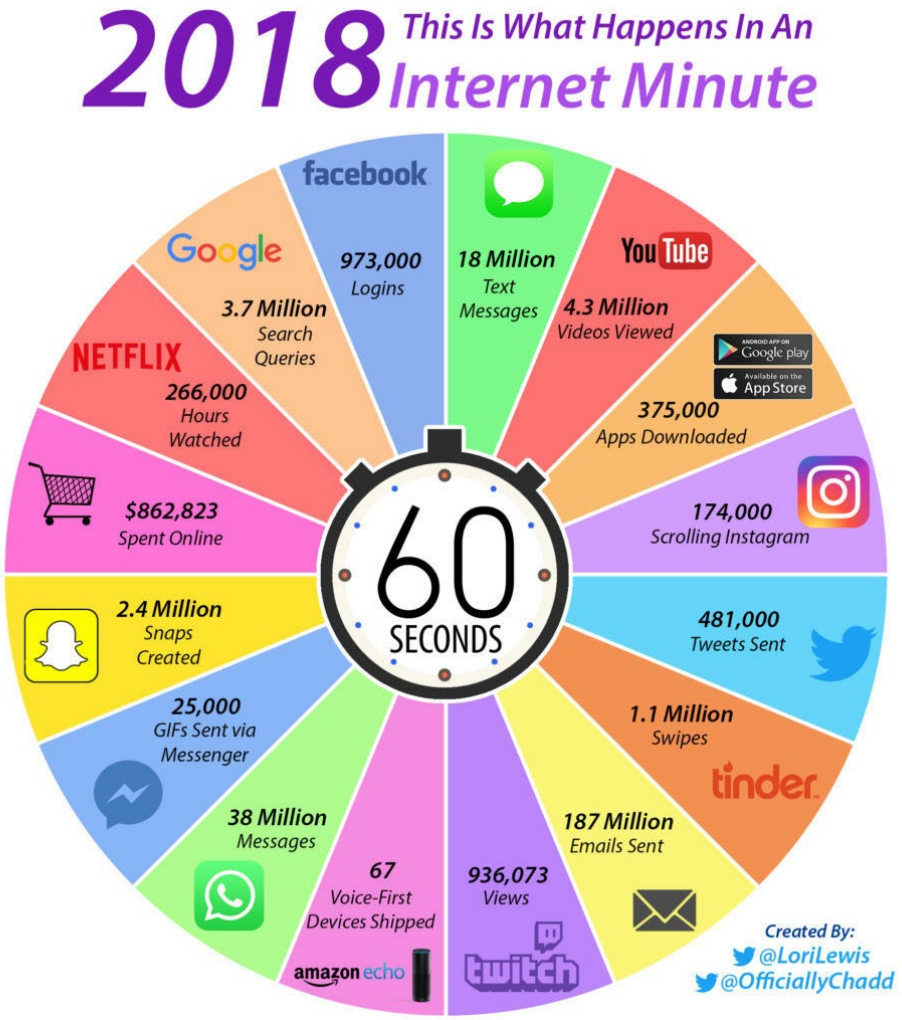
P ❑ *The number of emails correctly classified as needs/does not need attachment*

❑ *None of the above- it's not a machine learning case.*



# WHY NOW?

- Huge amount of data
- Computational power

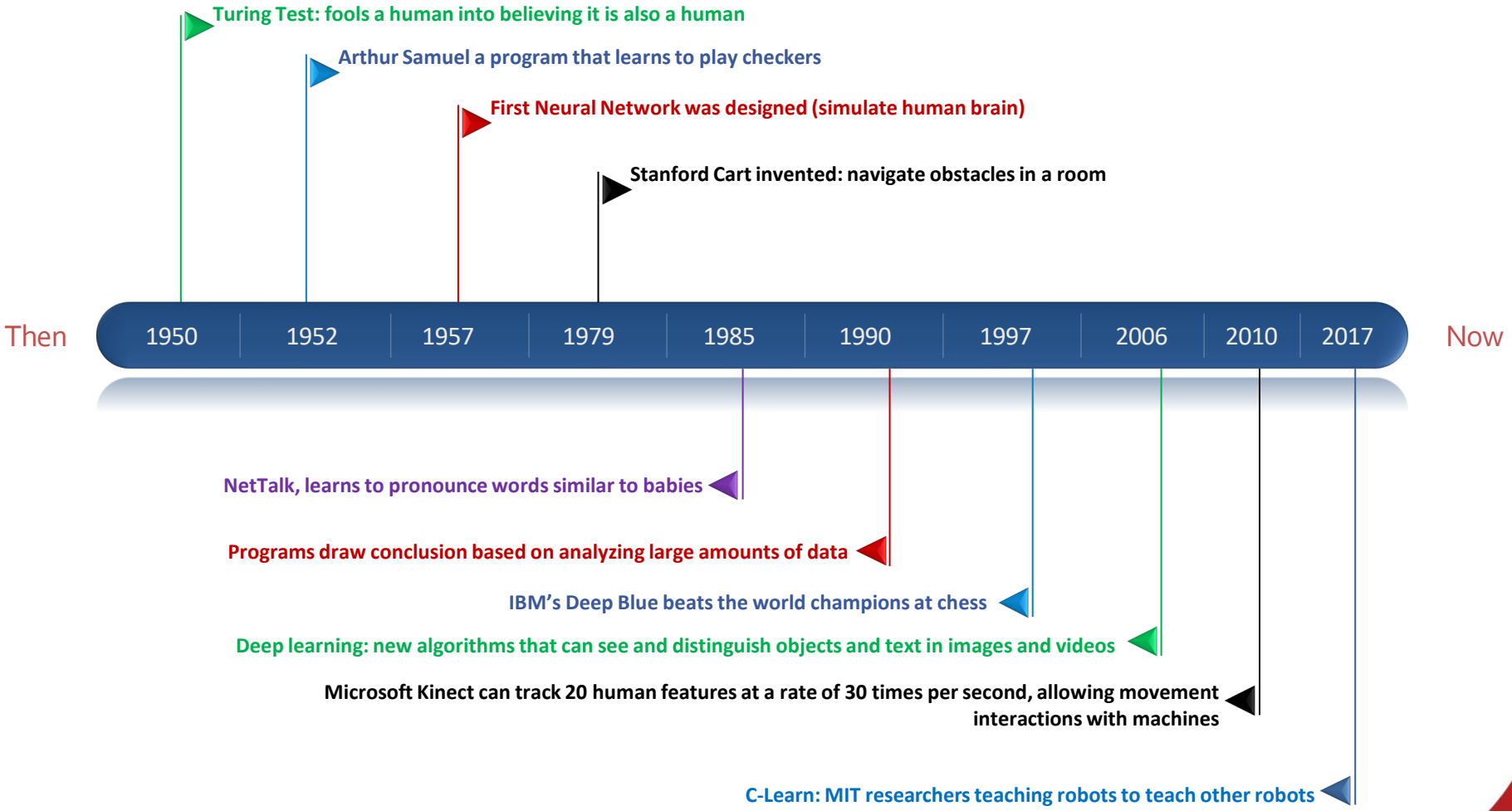


# TEXTUAL DATA

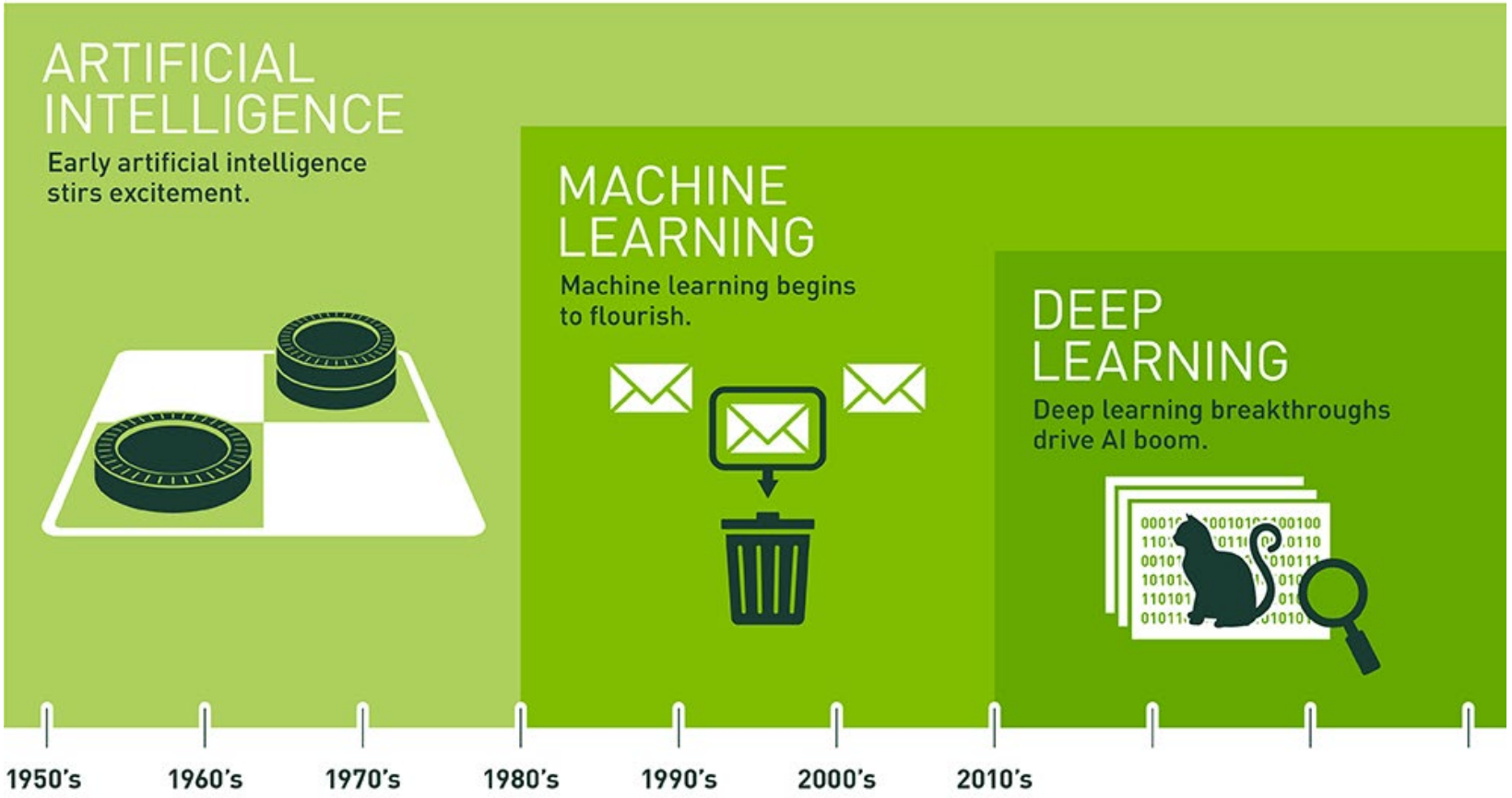
# Applications of Machine Learning

Machine learning is preferred approach to:
Speech recognition – Alexa, Siri, Google …etc
Face Recognition – Facebook tag a friend suggestion
Fraud Detection - PayPal
Travel Shortest path - Google Maps, Uber
Recommendations – Netflix, Spotify
Etc…….

# Machine Learning Timeline



# MACHINE LEARNING TIMELINE



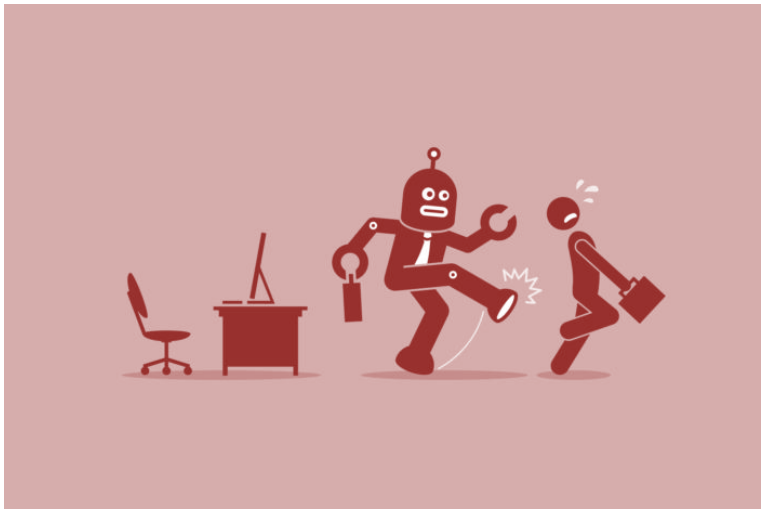
# DEMO

NetTALK
https://www.youtube.com/watch?v=gakJlr3GecE
Stanford Cart
https://www.youtube.com/watch?v=ypE64ZLwC5w
IBM Deep Blue Short Documentary
https://www.youtube.com/watch?v=NJarxpYyoFI

# Will Machine Learning Replace Us?

ML will unlikely be able to replace every human job.

But! Machines have been slowly replacing human workers since the industrial revolution- 18th century!



Lancaster University

# Major Jobs Replaced With Machine Learning

- Assembly-line and factory workers.
- Bus, taxi, and truck drivers. Tesla / Uber
- Phone operators, telemarketers and receptionists.
- Cashiers. Self-checkout lines
- Bank tellers and clerks. ATMs, digital currency i.e. Bitcoin
- Packing, stockroom and warehouse moving. Amazon
- Prescription. Pharmacy robots
- Auto Pilots, auto bartenders, postal machines, surgeon robots, soldiers, travel agents, ….etc

Lancaster University

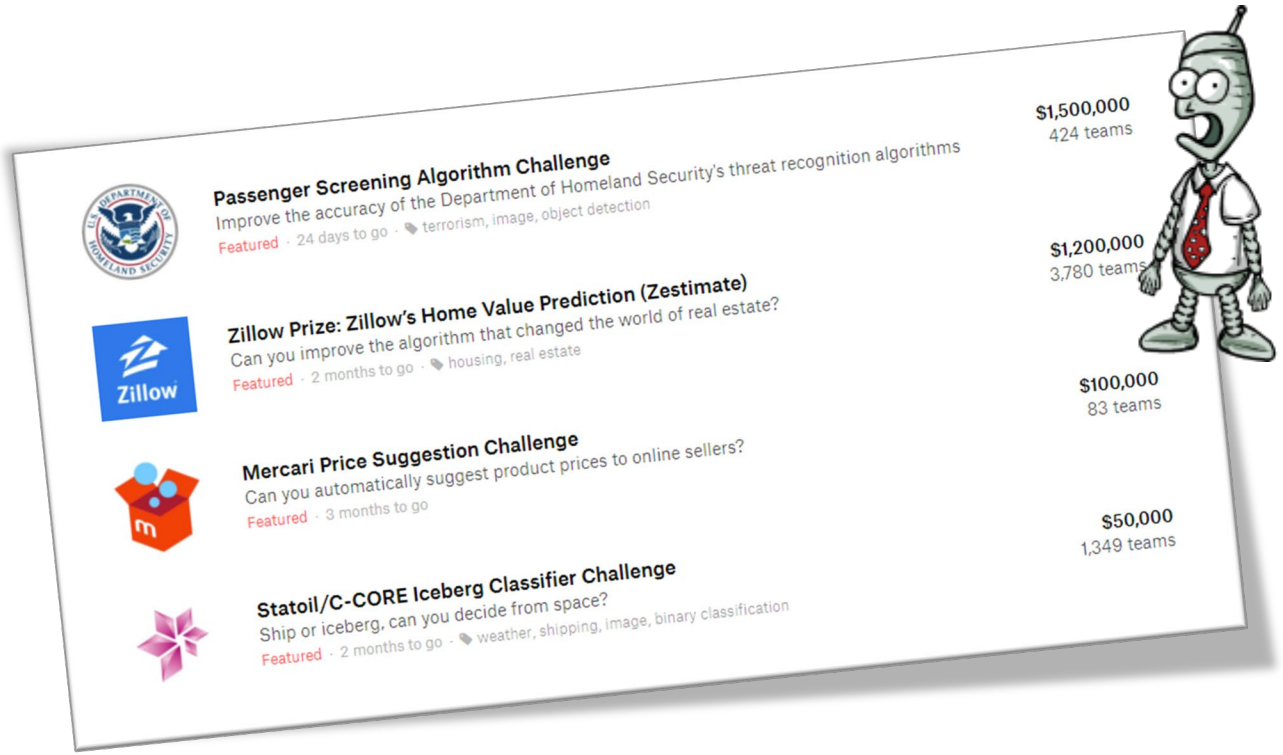
# MACHINE LEARNING PHILOSOPHY

- Limitations (relevant data, interpretation of results)
- Ethics: learning bias (high paid job ads for males)
- Discrimination: automatic decisions (credit scoring)

# Machine Learning Competitions

Kaggle Competitions: Challenge yourself with real-world machine learning problems.

https://www.kaggle.com/competitions



Lancaster University

# Part 1:
# Machine Learning Fundamentals

Learning Process

Aim

Training and Testing

Cross-validation and Overfitting

Learning Algorithms

Introduction to NB, SVM and RF

Features Extraction

# Learning Process

- The process of learning begins with :
  1. Observations / data (examples, direct experience, instructions)
  2. Finding patterns (e.g., those who buy cereal buy milk)
  3. Make better decisions (e.g., place milk near cereal, but away from seafood)

# Aim

- The primary aim is to allow the computers to learn (training) automatically and adjust actions accordingly to make decisions on future events (testing).

# TRAINING

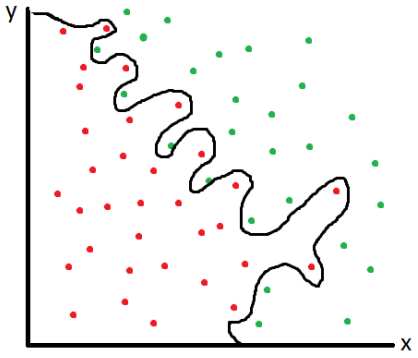
- Machine Learning Algorithms learn from data (training data).
- To find relationships, develop understanding, make decision and evaluate their confidence from the data given.
- The quality and quantity of the training data has as much to do with the success as the algorithms themselves.
- Training ends up with building a model.

# TESTING

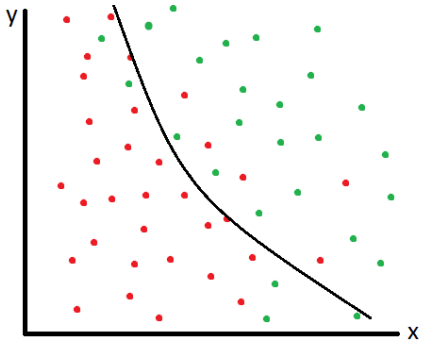
- To evaluate the trained model we use data that the model has never seen before (testing data)
- Usually the dataset is split into 90% training and 10% testing (or 70%, 30%) on different random folds (this is called cross validation).
- Cross validation helps avoiding overfitting.
- Best practice is to test your model on a testing dataset similar to the training one as otherwise you risk misclassification errors.

# OVERFITTING

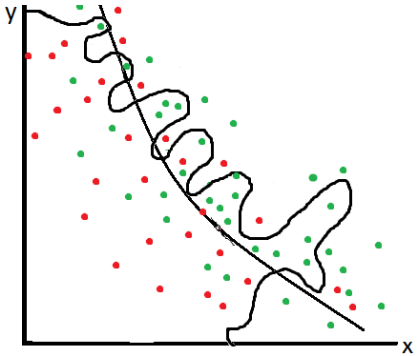
- Overfitting: your model fits with a particular training dataset but fails to fit additional data or predict future observations reliably.
- Avoid creating a model that is tailored only to your training data (e.g., assuming all fruit are round where training dataset includes only apples and oranges).
- ML works on predicting **y** instances given **x** instances:




Overfitting




Balanced




Unseen data

# Machine Learning Algorithms

**Algorithms** simply put, they learn from and make predictions on data.

Such algorithms overcome weaknesses of static (hard coded) program instructions by making data-driven predictions or decisions through building a model from sample inputs.

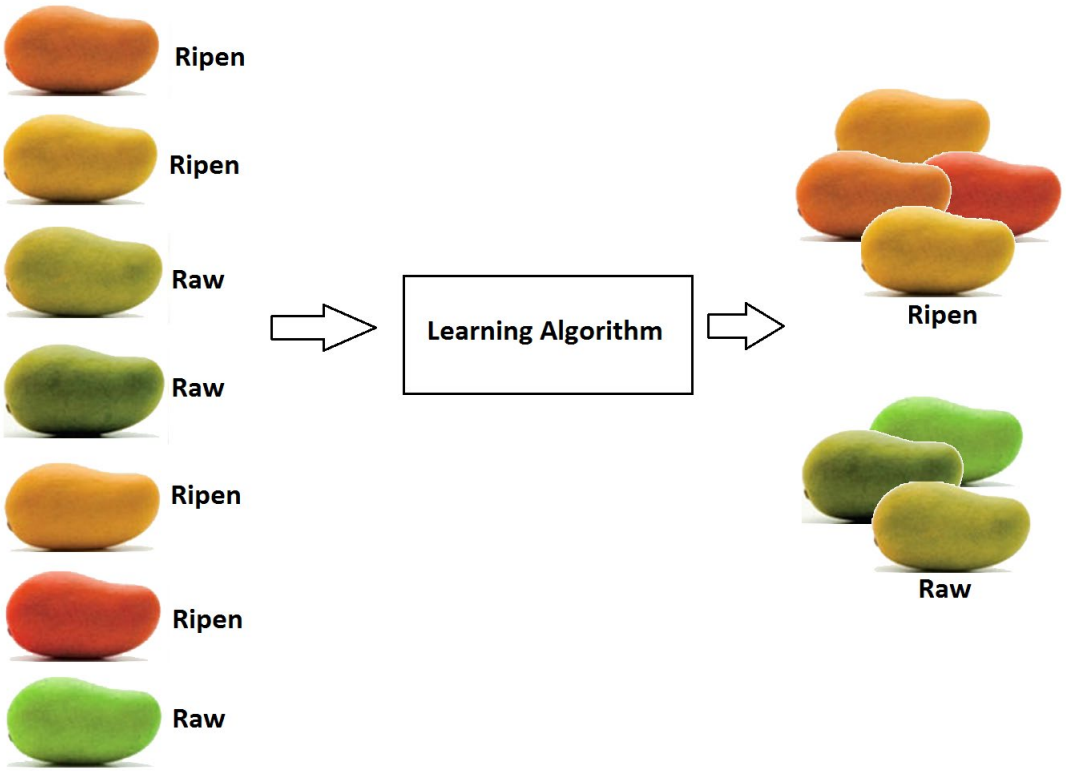
# Machine Learning Algorithms

- **Supervised**
  - Naïve Bayes, Decision Trees (Random Forest),…
- **Unsupervised**
  - Clustering: K-means.
- **Semi-supervised**
  - A combination of both

Note that both theory and empirical evidence provide little systematic guidance about which method is likely to dominate; performance depends on particular setting and dataset so NLP researchers often try multiple methods and then pick the best performer in the particular circumstances

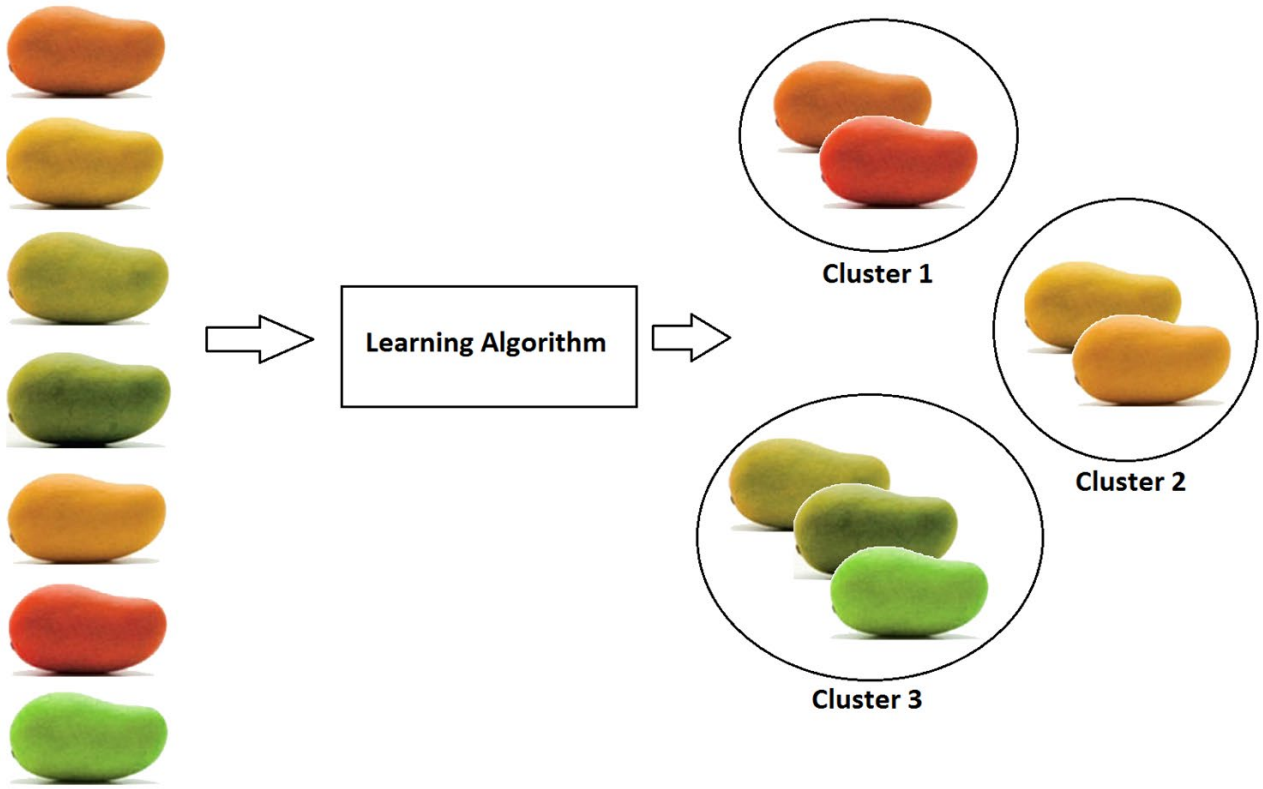
# Supervised Machine Learning

- Apply what learnt in the past to new data
- Using labelled examples to predict future events



# Unsupervised Machine Learning

- Training data is unlabelled.
- Describe a hidden structure from unlabelled data.



# Semi-supervised Machine Learning

Lancaster University

- Falls somewhere between supervised and unsupervised
- Using both labelled and unlabelled data
- Why semi?

  Acquiring labelled data is costly.

# Features and Features-Extraction

- Feature: an individual measurable property or characteristic of a phenomenon being observed.

- Shape, colour, texture, taste, price, origin…

# Features Reduction

- Having irrelevant features in your data can decrease the accuracy of many models.

- **Reduces Overfitting**: Less redundant data means less opportunity to make decisions based on noise.
- **Improves Accuracy**: Less misleading data means modelling accuracy improves.
- **Reduces Training Time**: Less data means that algorithms train faster.

# NAÏVE BAYES (NB)

Lancaster University

Set of algorithms based on Bayes Theorem (**probability** of an event based on knowledge of conditions that might be related to the event). P(Play|Sunny, clear sky)

**Why naïve?** Because it treats features independently.

A fruit may be considered to be an apple if it is **red**, **round**, and about 3" in **diameter**.

NB classifier considers each of those "features" to contribute independently to the probability that the fruit is an apple, regardless of any correlations between features. Which is not always the case.

P(Apple|red, round, 3"diam)

# NAÏVE BAYES (NB)

Lancaster University

**Advantages**

- It's relatively simple to understand and build
- It's easily trained, even with a small dataset
- It's fast!
- It's not sensitive to irrelevant features (handles missing values)

**Disadvantages**

- It assumes every feature is independent, which isn't always the case (rain vs clouds).

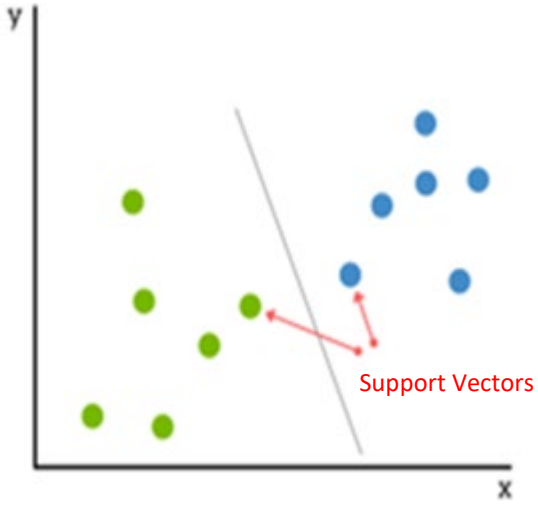
# SUPPORT VECTOR MACHINE (SVM)

Supervised classifier. Finds a **hyperplane** that best divides a dataset into two classes.

Think of a hyperplane as a line that linearly separates and classifies a set of data.

Support vectors are the data points nearest to the hyperplane.

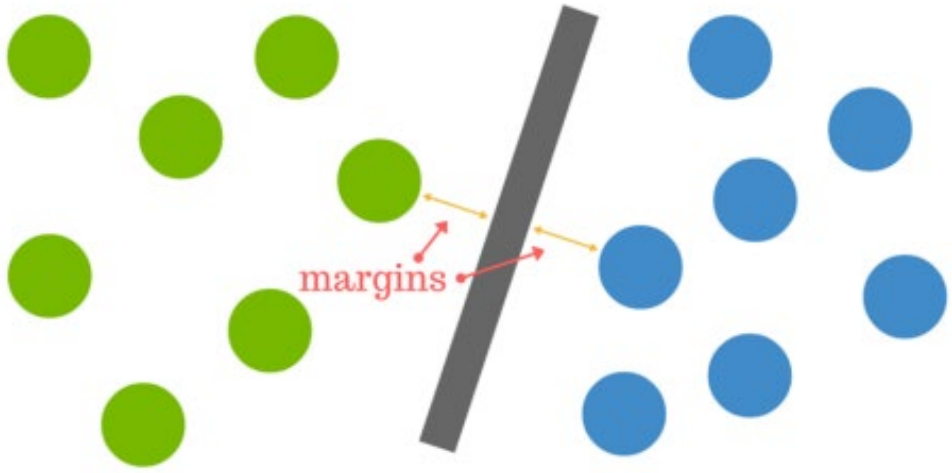
Points that are close to the opposing class.



# SUPPORT VECTOR MACHINE (SVM)

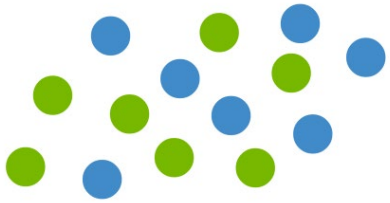
The distance between the hyperplane and the nearest data point from either set is known as the **margin**.

The goal is to choose a hyperplane with the greatest possible margin between the hyperplane and any point within the training set.

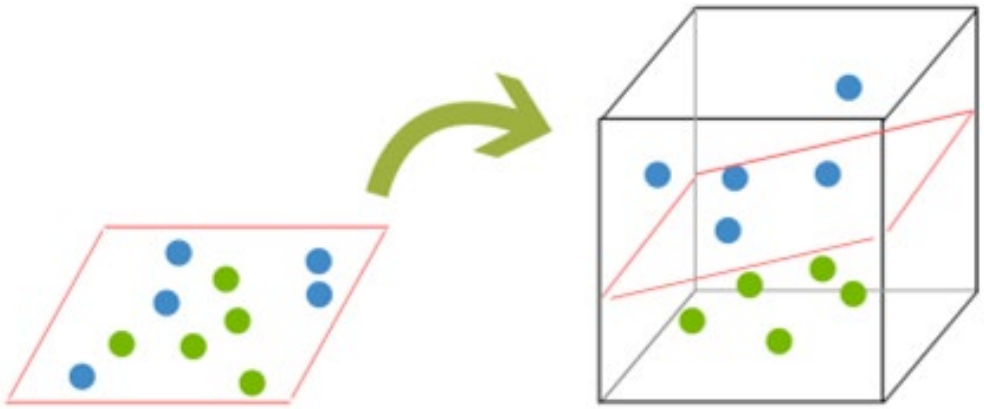


# Support Vector Machine (SVM)

But what happens when there is no clear hyperplane?



In order to classify a dataset like the one above it's necessary to move away from a 2d view of the data to a 3d view



# Support Vector Machine (SVM)

Lancaster University

**Advantages**

- Accuracy
- Works well on smaller cleaner datasets

**Disadvantages**

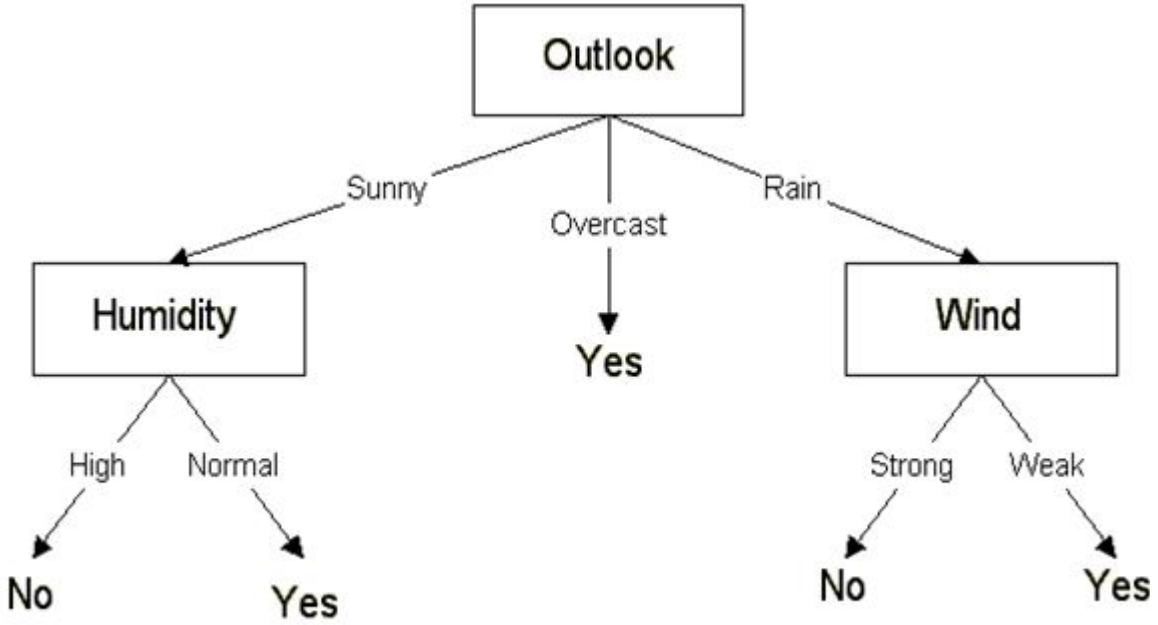
- Not suited to larger datasets as the training time with SVMs can be high
- Less effective on noisier datasets with overlapping classes
- Requires you to handle missing data.

# Random Forest (RF)

- Very popular.
- Also known as random decision forest, it is a ensemble method.
- Ensemble means it uses multiple learning models to gain better predictive results.
- Therefore, creating an entire ‘forest’ of random uncorrelated decision trees to arrive at the best possible answer.

# RANDOM FOREST (RF)

- Top-down approach where the root nodes creates binary splits until a certain criteria is met.
- The more trees in the forest the more robust is the prediction and thus high accuracy.



# Random Forest (RF)

**Advantages**

- Robust with high accuracy
- Handles missing values.
- Handles small and large datasets

**Disadvantages**

- Overfitting in noisy training data (poor performance on unseen data).
- Very vulnerable to changes in testing and training data

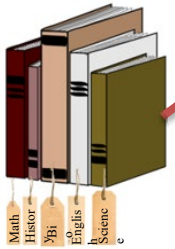
# How much training data is required

- The quality and amount of training data is often the single most dominant factor that determines the performance of a model.
- But exactly how much training data do you need? The correct answer is: it depends …. on:
  - task
  - performance to achieve
  - input features
  - noise in data
  - noise in features
  - and complexity of the model (number of classes).

# SUMMARY OF PART 2

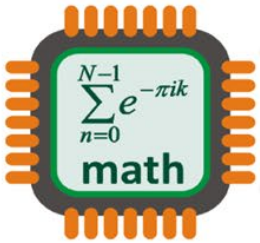
Unsupervised

Supervised




Feature Extraction

ML Algorithm




Clustering

Predictive Model

Unseen Data


Classification

Lancaster University

# Part 2:
# Text Classification

Introduction to Text Classification
Textual Features
Practical Text Classification Example

# What is Text Classification

- Text Classification assigns one or more classes to a sample of text (e.g., sentence, document, etc.) according to its content.

- Examples:
  - Detecting spam emails (Spam / Not Spam)
  - Detecting news articles topics (Politics, Sports, Science, ….)
  - Assist search engines (Relevant results on Google/Bing search engines)

- Examples in Acc&Fin:
  - Determining the sentiment of a sentence or media article
  - Identifying potential fraud cases

# Features for Textual Data

- Verbs, Nouns, Pronouns
- Adverbs, Function words
- Word length
- Number of syllables
- Topic
- Words frequency
- Different words
- Syntax, Semantic
- Readability
- ….etc

# FEATURES FOR TEXTUAL DATA

## Chairman's Statement

**Chairman**'s statement: This is **my** first report to you as **Chairman** of your board of directors, following the successful admission of the **Group**'s shares to trading on AIM on 10 January 2008. 1700 **Group** Plc was incorporated on 9 May 2007 but did not trade until the acquisition of its first trading subsidiary, Hamblyn Selection Ltd ("Hamblyn") on 14 November 2007. The subsequent acquisition of the **Group**'s second trading subsidiary, Inspired Selection Ltd ("Inspired"), was completed on 10 January 2008. **I** would like to express **my** thanks to the **Group**'s employees

## Governance Report

Corporate **governance** statement: Statement of compliance with the **Combined Code**. The **Group** recognises the value of the Principles of Good **Governance** and The **Combined Code** on Corporate **Governance** published in 2006 by the Financial Reporting Council ('the **Combined Code**'). The **Group** intends to comply with the **Combined Code** so far as is practical and appropriate for a public **group** of its size and nature. The **Group** supports the recommendations on corporate **governance** of the Quoted Companies Alliance (QCA) and has implemented steps to reach compliance.

Lancaster University

# Part 3:
# Deep Learning (Neural Network)

Introduction to Deep Learning

# Deep Learning


"Now! … *That* should clear up a few things around here!"

# Deep Learning

- is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called **artificial neural networks**.

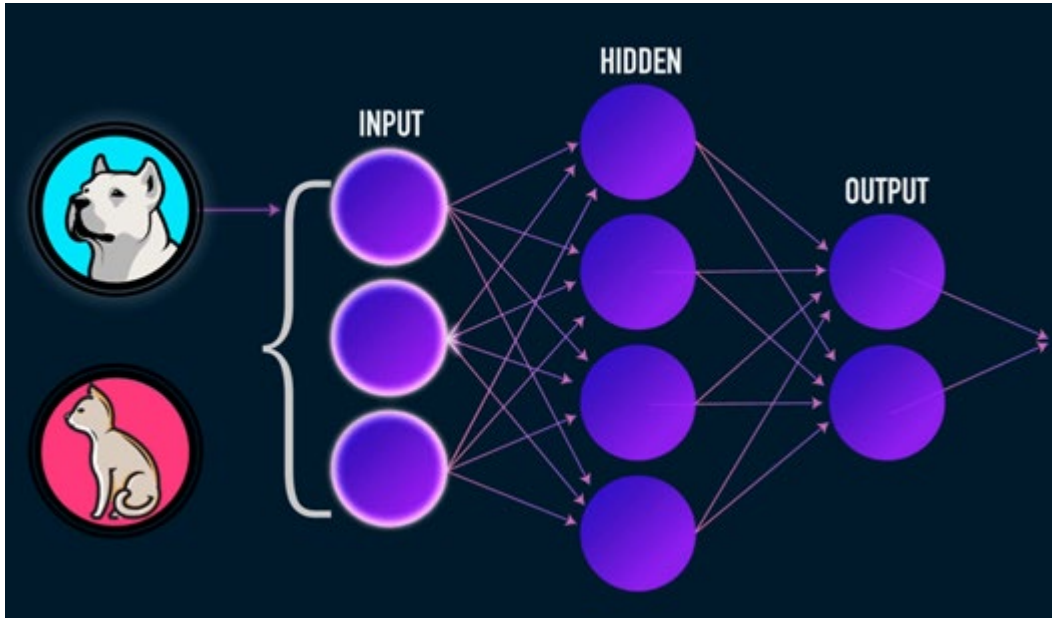
- Generally speaking, deep learning is a machine learning method that takes in an input **X**, and uses it to predict an output of **Y**.

# How does it work?

- Given a large dataset of input and output pairs,
- a deep learning algorithm will try to minimize the difference between its prediction and expected output.
- By doing this, it tries to learn the association/pattern between given inputs and outputs
- this in turn allows a deep learning model to generalize to inputs that it hasn't seen before.

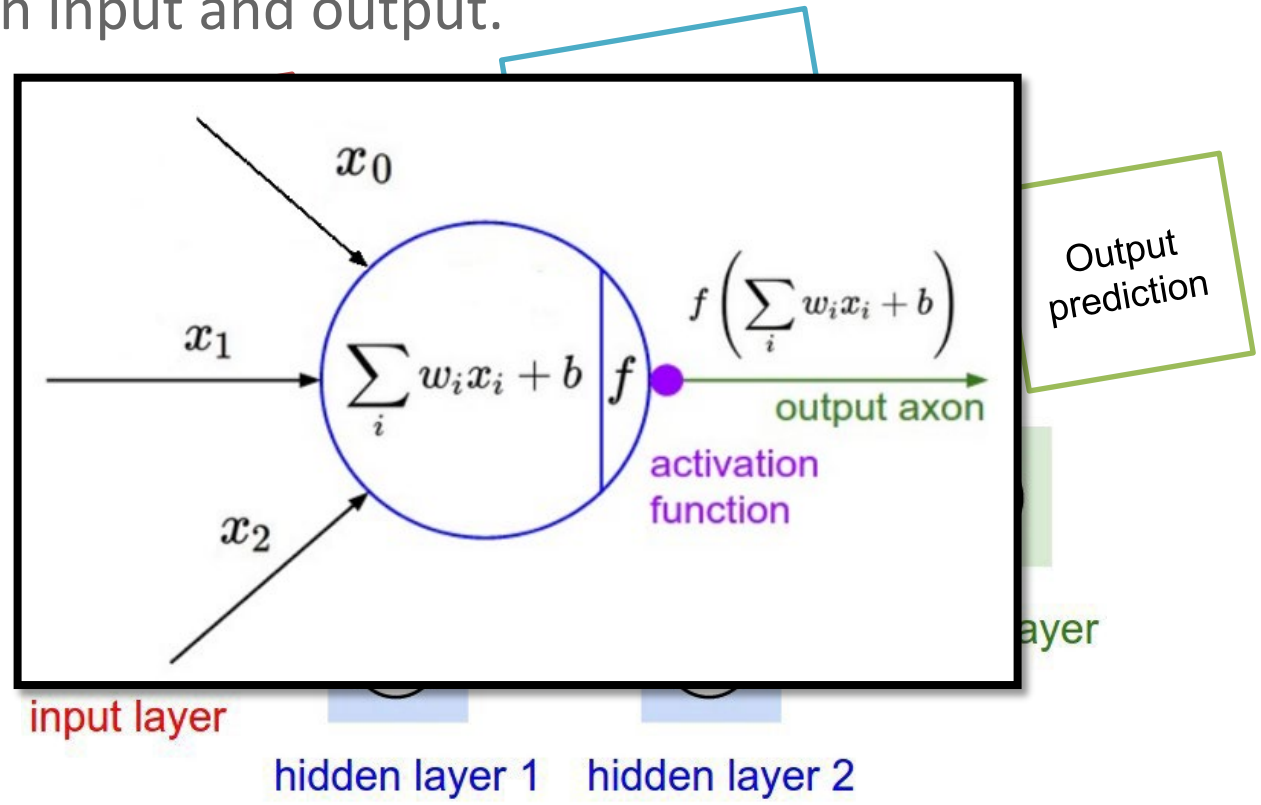
# EXAMPLE

If image is labelled as dog but DL predicts cat, then DL learns the features of the given image associated with dog.



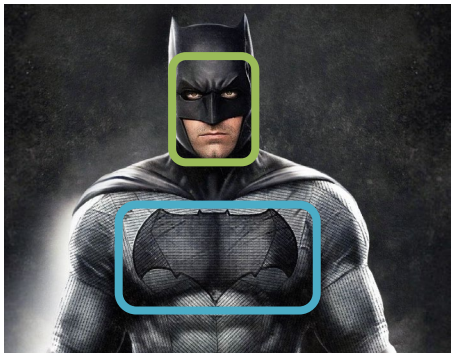
# How Do Deep Learning algorithms "learn"?

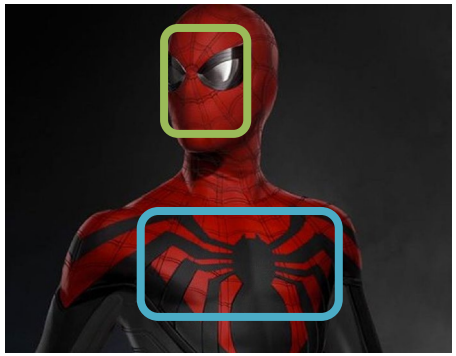
- DL algorithms use Neural Network to find associations between input and output.

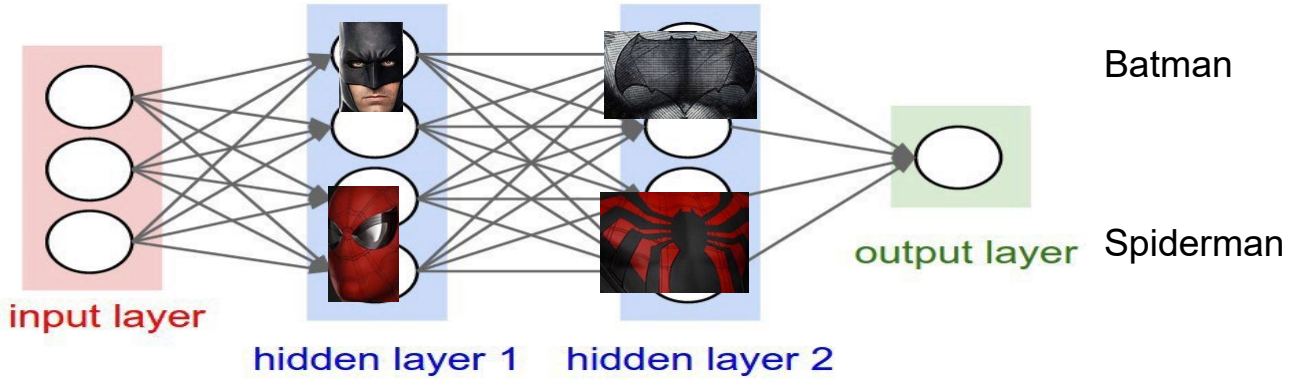


# WHAT IS A HIDDEN LAYER?

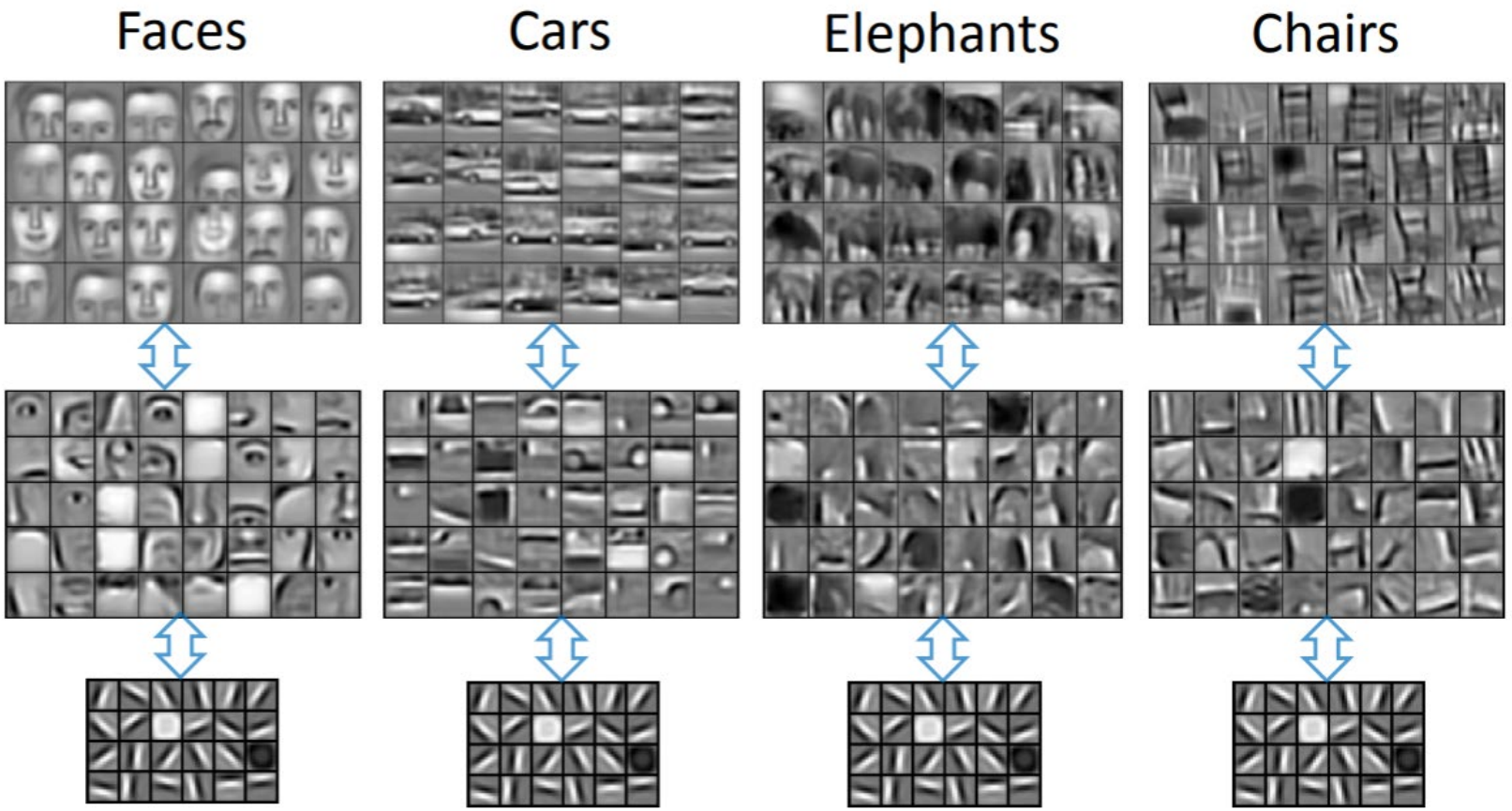
- artificial neurons take in a set of weighted inputs and produce an output through an activation function





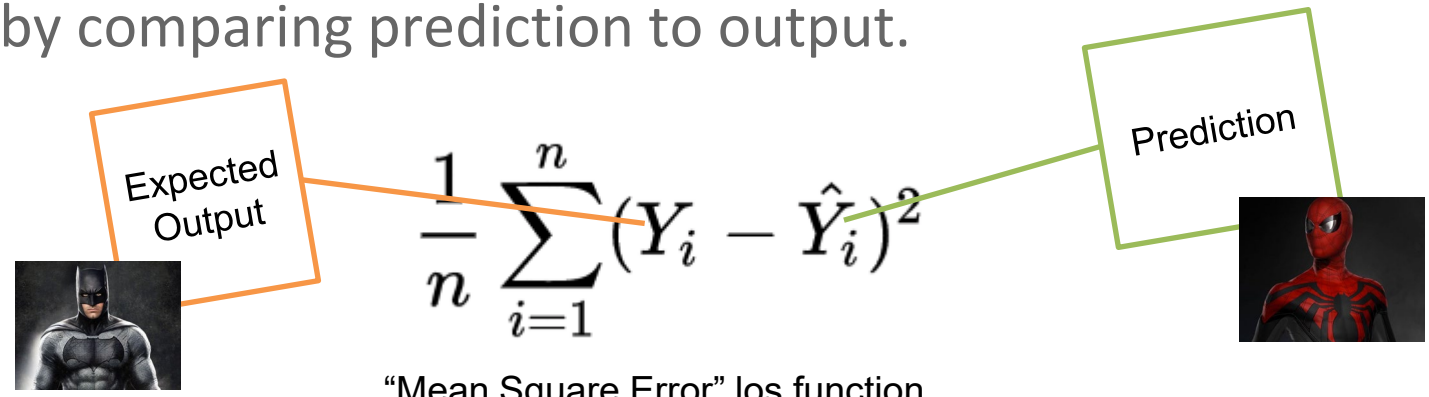


# Another Example



# Loss Function

- After the DL NN passes its inputs all the way to its outputs,
- the network evaluates how good its prediction was
- by comparing prediction to output.




"Mean Square Error" los function

# Deep Learning Goal

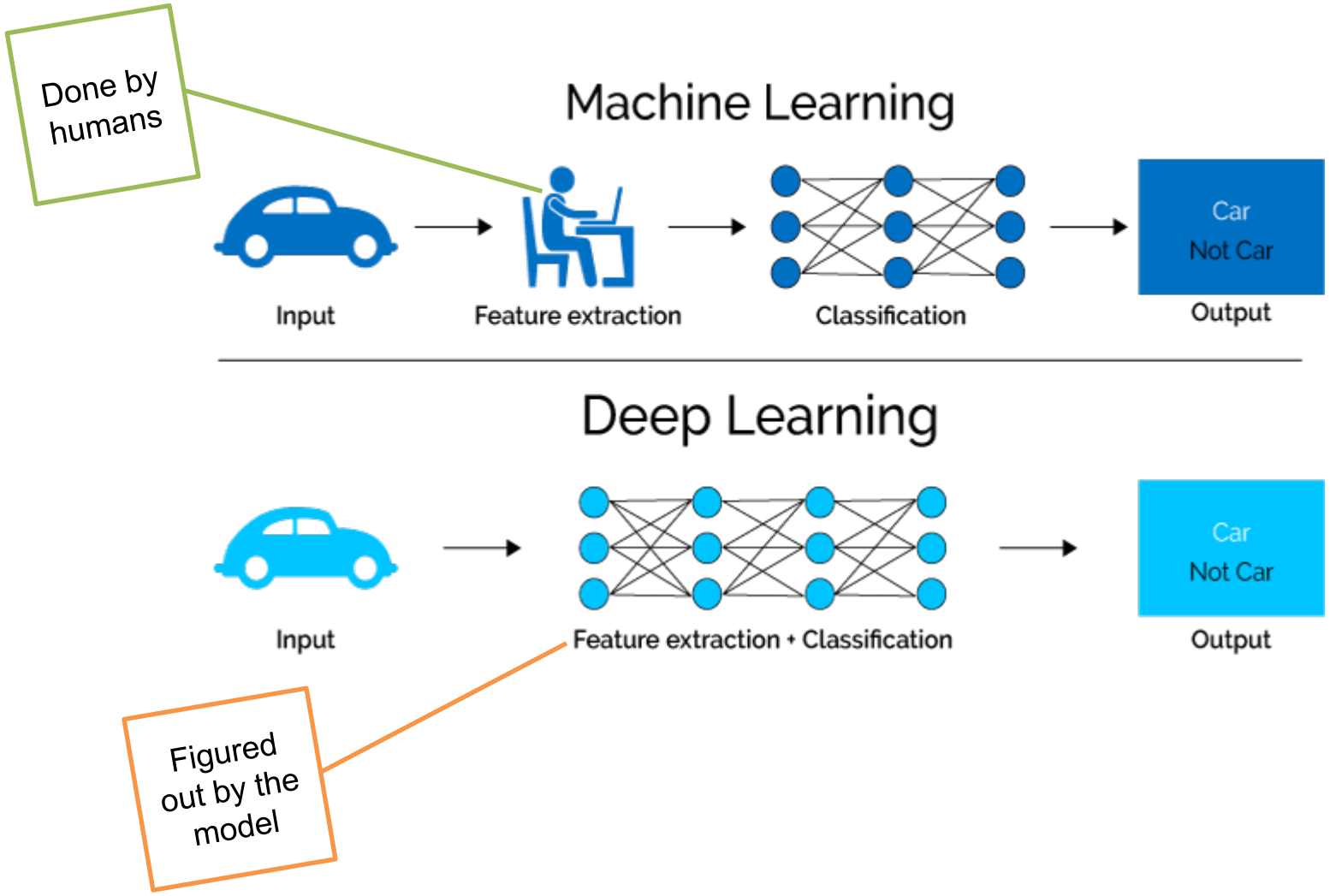
Lancaster University

- Minimize Loss by adjusting weights of the network.
- Using something called “Back propagation”
- the network backtracks through all its layers to update the weights of every node in the opposite direction of the loss function
- in other words, every iteration of back propagation should result in a smaller loss function than before.
- the continuous updates of the weights of the network ultimately turns it into a precise function approximator — one that models the relationship between inputs and expected outputs.

# So why is it called "Deep" Learning?

- The "deep" part of deep learning refers to creating deep neural networks.
- This refers a neural network with a large amount of layers — with the addition of more weights, the neural network improves its ability to approximate more complex functions.
-

# Deep Learning vs Traditional Machine Learning



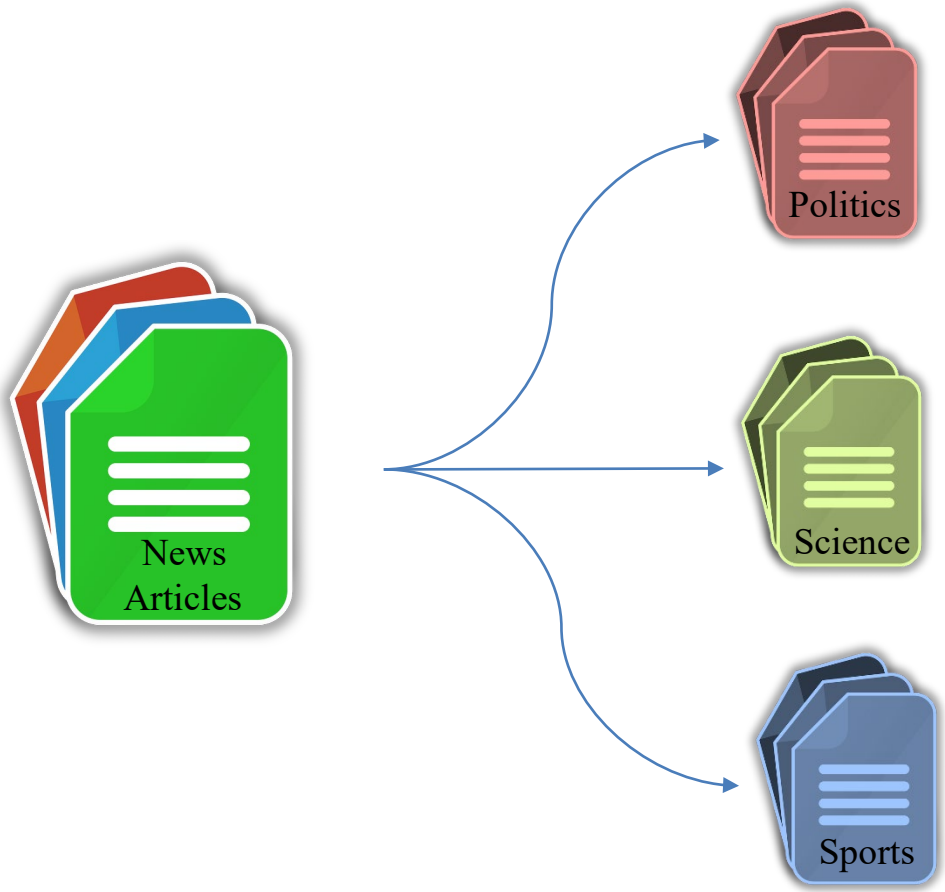
# Deep Learning vs Traditional Machine Learning

- Deep learning works better with more data (the larger the better).
- With Small data traditional machine learning algorithms are preferred.
- Deep learning requires high computational time and high end infrastructure (GPU).
- When there is lack of domain understanding for features selection Deep Learning outshine (worry less about features).
- Deep learning works well with complex problems (image classification, NLP, speech recognition ...etc).

# CONCLUSION

- Deep learning is ultimately an expansive field, and is far more complex than I've described it to be.
- Various types of neural networks exist for different tasks (e.g. Convolutional NN for computer vision, Recurrent NN for NLP),
- and go far and beyond the basic neural network that I've covered.

# PRACTICAL: CLASSIFYING DOCUMENTS



# THE TASK

- Building a model using **WEKA** to classify Wikipedia Articles into either Spanish or English articles. So basically detecting the language of the articles.

# DATASET

- 70 Spanish Wikipedia articles
- 70 English Wikipedia articles

- Articles saved in plain text file formats ending with .txt.

# Machine Algorithm Tool

- For the purpose of this workshop we will be using WEKA machine learning algorithms to classify our data.

# WHAT IS WEKA?


Lancaster University

- Weka is a collection of machine learning algorithms for data mining tasks.
- Weka contains tools for:
  - data pre-processing,
  - **classification**,
  - regression,
  - clustering,
  - and visualization

# WEKA JAVA

- WEKA is originally written in Java but in this workshop we will be using the Graphical User Interface (GUI) application.

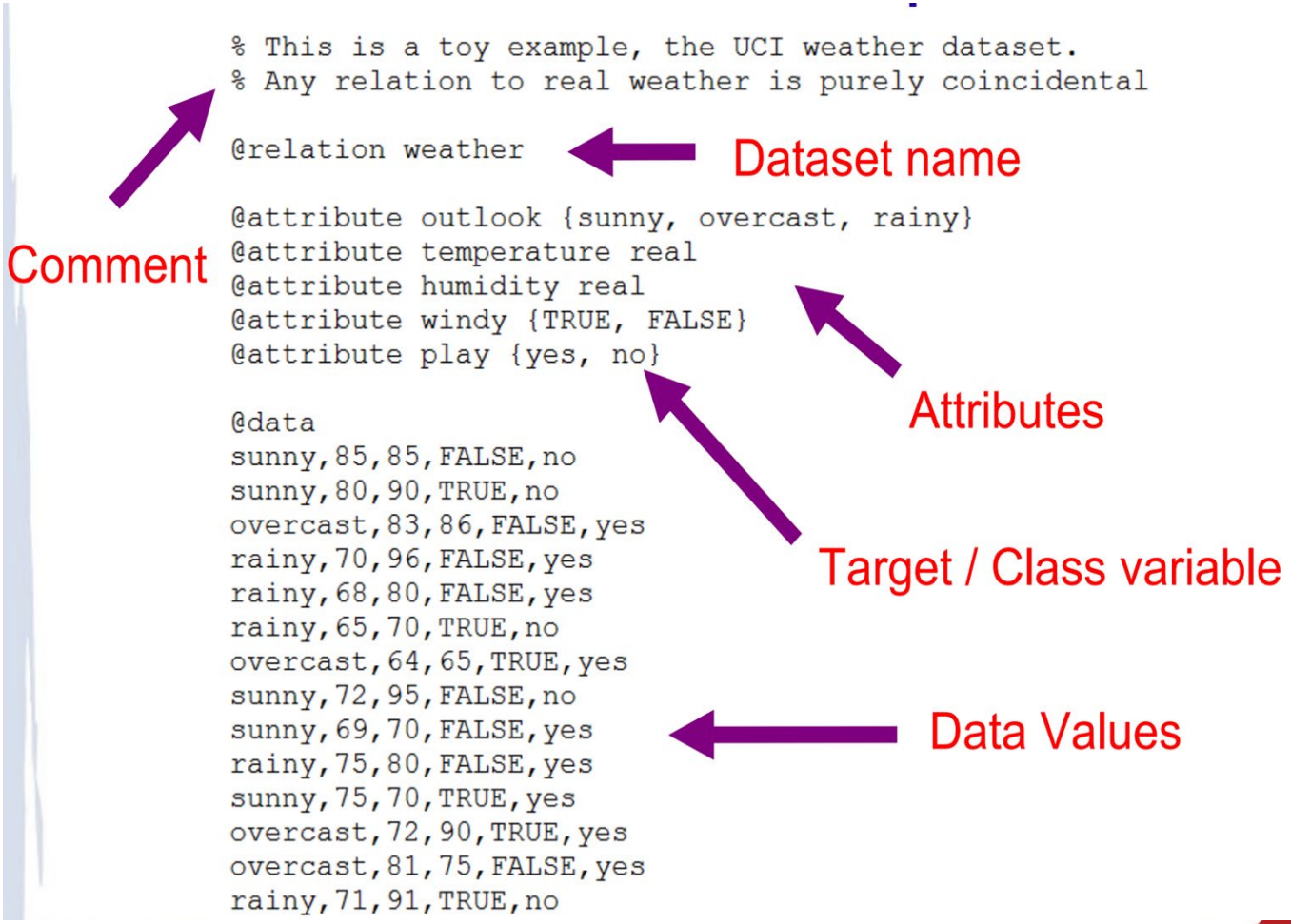
- So no coding required.

# USING WEKA TO CLASSIFY TEXT

- - Classify text into two classes
- - Compare NB vs RF vs SVM
- - Reduce attributes (features)

# READING DATASET INTO ARFF

- ARFF: Attribute-Relation File Format
- Used for WEKA
- Structured format



# Confusion Matrix (for Classifications)

Lancaster University

| Confusion Matrix | | True Condition | |
|---|---|---|---|
| | | **Actual +ve** | **Actual -ve** |
| **Predicted Condition** | **Pred. +ve** | TP | FP (Type I) |
| | **Pred. -ve** | FN (Type II) | TN |

- TP (or TN) Rate = Percentage of True Positives (Negatives)
  - Aka recall (next slide), TPR = probability of detection
  - TPR = sensitivity, TNR = specificity
  - TPR = TP/(TP+FN) ; TNR = TN/(TN+FP)

- FP Rate = Percentage of False Positives (FN similarly for Negatives)
  - Aka fallout, FPR = probability of a false alarm
  - FPR = FP/(FP+TN)

# NLP Measures[1]

[1]: Powers, D.M.W; Evaluation: From Precision, Recall, and F-Measure to ROC, Informedness, Markedness & Correlation, JMLT, 2011

$$recall = \frac{Number\ of\ document\ retrieved\ that\ are\ relevant}{Total\ number\ of\ documents\ that\ are\ relevant}$$

i.e. what proportion of actual positives were identified correctly?
aka – sensitivity, same measure as True Positive Rate

$$precision = \frac{Number\ of\ document\ retrieved\ that\ are\ relevant}{Total\ number\ of\ documents\ that\ are\ retrieved}$$

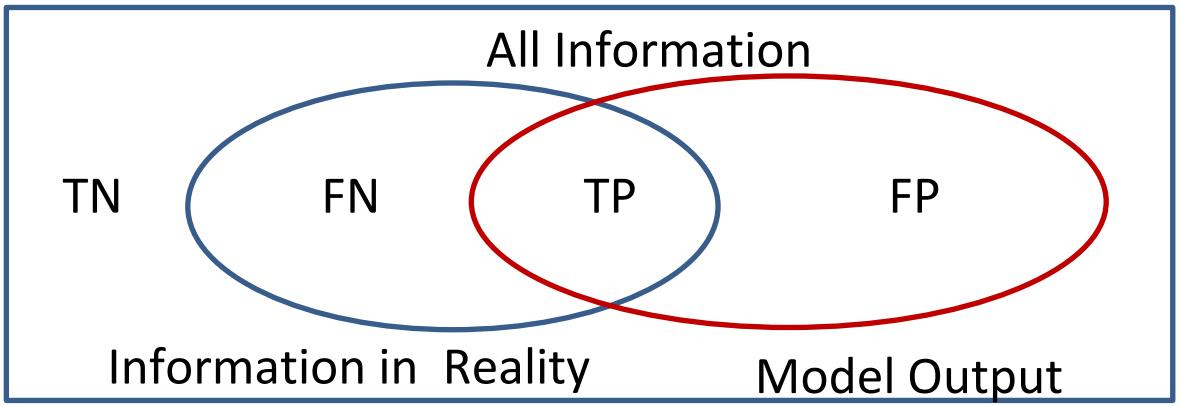
i.e. what proportion of positive identifications were actually correct?

$$accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}$$

i.e. what proportion of predictions were actually correct?

$$F\ measure = \frac{2\ x\ recall\ x\ precision}{recall + precision}$$

# NLP Formula




- TP = True Positive
- FP = False Positive
- TN = True Negative
- FN = False Negative

- Accuracy = (|TP|+|TN|) / (|TP|+|TN|+|FP|+|FN|) used in Lift charts
- Recall = |TP| / ( |TP| + |FN|) used in ROC curves[2]

[2]: Receiver Operating Characteristic Curves, plots TPR vs FPR

- Precision = |TP| / (|TP| + |FP|)

Lancaster University

# Text Classification Practical

## WEKA Classifier

# NEEDED SOFTWARE

1 – WEKA

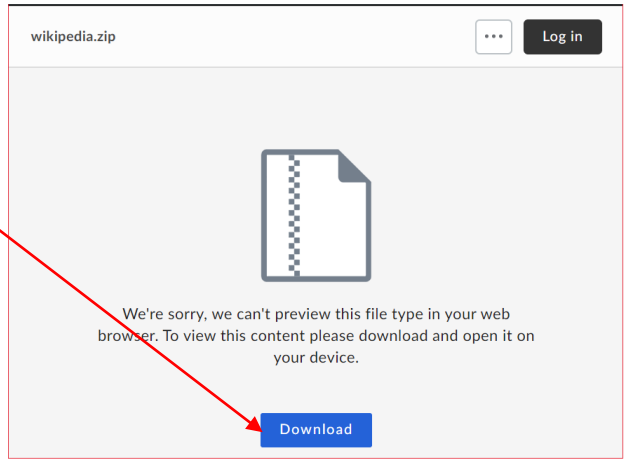
2 – NOTEPAD ++

# Needed Materials

- Create a new folder called it **classification** on your desktop:

  C:\Users\username\Desktop\classification

- Go to http://bit.ly/2XPZ0RI and download Wikipedia.zip (click Download):




- Downloaded as .zip files. Right click and extract all contents.

# HANDS-ON TASK STEPS

1. **Convert dataset into ARFF file format.**
2. Prepare dataset and extract features
3. Train a classifier using training data
4. Create a classification model
5. Test the classification model

# CONVERT DATASET INTO ARFF FILE FORMAT

- Our data comes in as text files (folder1: EN, folder2: ES) with each containing 70 text (.txt) files


EN-madrid-ES-EN-1.txt
EN-madrid-ES-EN-2.txt
EN-madrid-ES-EN-3.txt
EN-madrid-ES-EN-4.txt

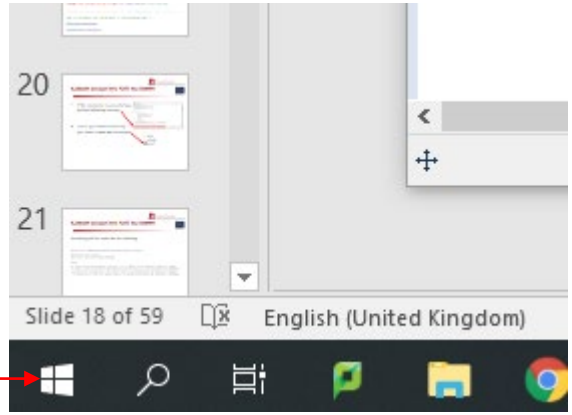
- We want to automatically convert the dataset into ARFF using a single command line.

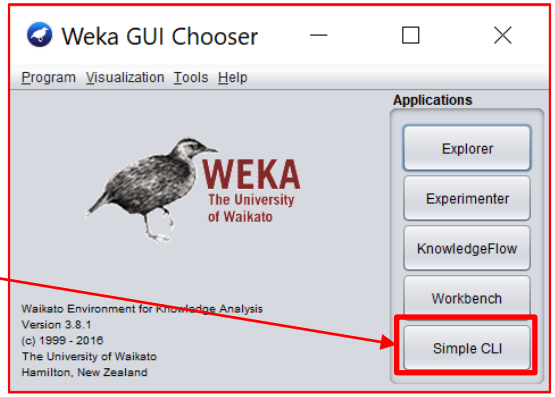

ES-madrid-ES-EN-1.txt
ES-madrid-ES-EN-2.txt
ES-madrid-ES-EN-3.txt
ES-madrid-ES-EN-4.txt

- To do this start by running WEKA GUI Chooser

# CONVERT DATASET INTO ARFF FILE FORMAT

- To do this start by running WEKA GUI Chooser
- If you can't find it, click Windows and type WEKA
- Click on Simple CLI



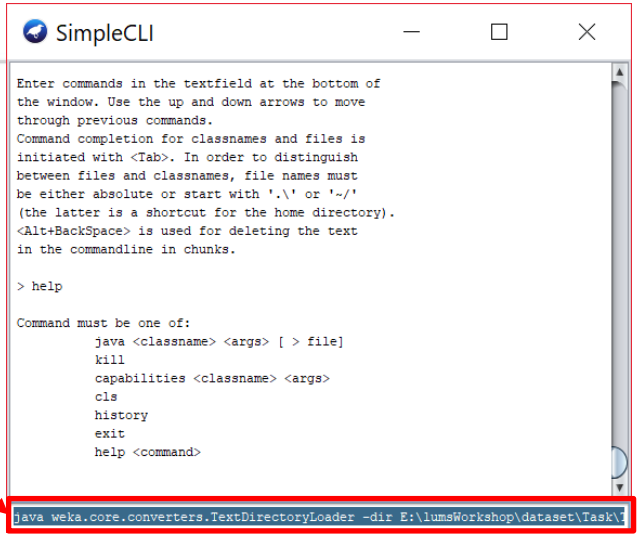



wiki.arff

Lancaster University

# CONVERT DATASET INTO ARFF FILE FORMAT

- Type following command in the command box below and hit enter:



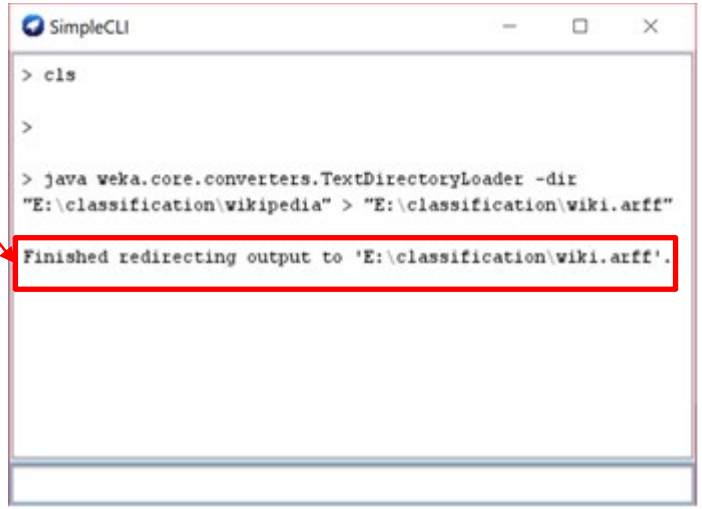

**java weka.core.converters.TextDirectoryLoader -dir Training_Directory > output_Arff_File**

**java weka.core.converters.TextDirectoryLoader -dir**
**C:\Users\username\Desktop\classification\wikipedia\>**
**C:\Users\username\Desktop\classification\wiki.arff**

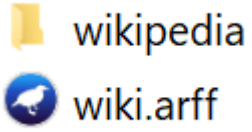
**Note: You may need to "skip" the backslashes "\" by adding another one in "\\".**
**Note: if your path has spaces you may need to surround it by double quotations e.g. "C:/my path/"**
**E.g.:**
\\lancs\homes\24\elhaj\Downloads\classification\wikipedia
becomes
\\\\lancs\\homes\\24\\elhaj\\Downloads\\classification\\wikipedia

# CONVERT DATASET INTO ARFF FILE FORMAT

- If the command is successful you get the following message




- And in your dataset directory you'll find a **wiki.arff** file created



# CONVERT DATASET INTO ARFF FILE FORMAT

Lancaster University

Resulting arff file looks like the following

```
@relation E__classification_wikipedia

@attribute text string
@attribute @@class@@ {EN,ES}

@data

'THE AMERICA MUSEUM\r\nThe America museum is a national Spanish museum ',EN
'MUSEO DE AMÉRICA\rEl Museo de América es un Museo Nacional español con sede en Madrid',ES

```

# CONVERT DATASET INTO ARFF FILE FORMAT

Lancaster University

But what about **testing** dataset?

WEKA is very strict with arff files created for training and testing and requires both to have the same format, structure and attributes.

For example if your training dataset has features such as colour, shape and size but your testing has only colour and shape, WEKA will consider them mismatching.

# CONVERT DATASET INTO ARFF FILE FORMAT

Lancaster University

- Best and easy way to create an arff file for your testing (unseen) dataset is by taking that out of the training arff file before you do any further work on it.

- For example our arff file has 70 ES and 70 EN instances.
- We'll take a copy of the file keeping only 5 instances for each class.
- We'll then replace each ES and EN with "**?**"

- User notepad++ to open and edit files.

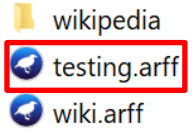
# CONVERT DATASET INTO ARFF FILE FORMAT

```
@relation E__classification_wikipedia

@attribute text string
@attribute @@class@@ {EN,ES}

@data

'space reserved for the museological purposes for which it was designed.\r\n',EN
' espacio para los fines museológicos para los que había sido concebido.\r\r',ES
'd his business with parks in other continents, as it was Six Flags\'s case.',EN
'su negocio con parques en otros continentes, como era el caso de Six Flags.',ES
```

Convert known labels to unknown (?)

```
@relation E__classification_wikipedia

@attribute text string
@attribute @@class@@ {EN,ES}

@data

'space reserved for the museological purposes for which it was designed.\r\n',?
' espacio para los fines museológicos para los que había sido concebido.\r\r',?
'd his business with parks in other continents, as it was Six Flags\'s case.',?
'su negocio con parques en otros continentes, como era el caso de Six Flags.',?
```

- Save the arff copy file as **testing.arff** in the dataset directory.
- We'll come back to this file later in the session.

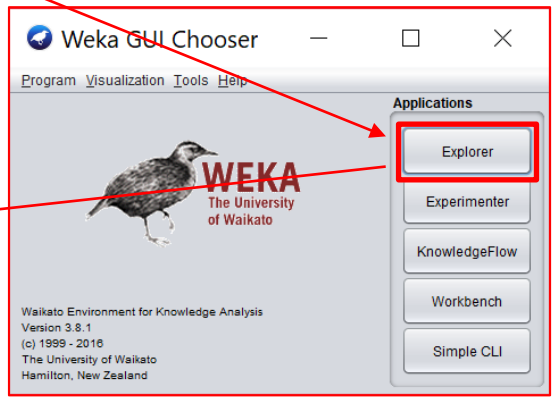


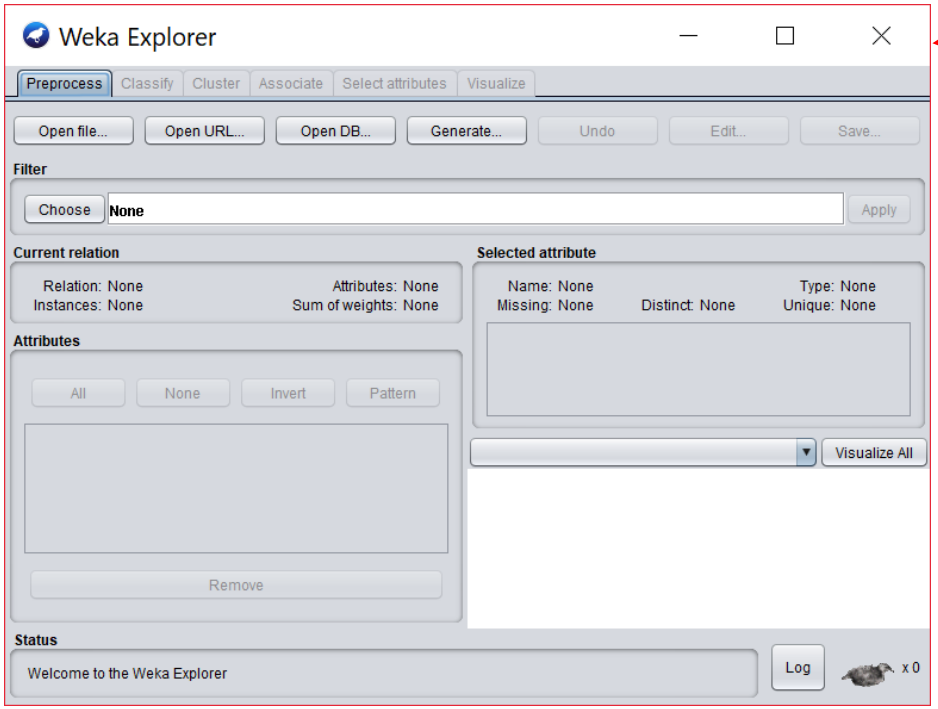
# Steps

1. Convert dataset into ARFF file format.
2. **Prepare dataset and extract features**
3. Train a classifier using training data
4. Create a classification model
5. Test the classification model

# PREPARE DATASET AND EXTRACT FEATURES

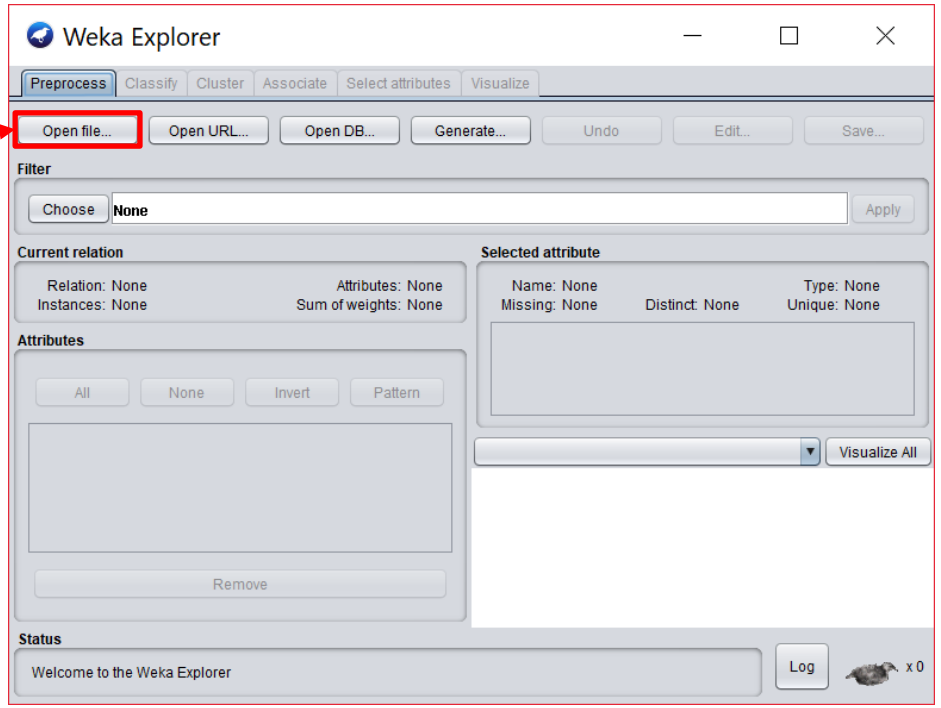
- In WEKA GUI Chooser click on “Explorer”

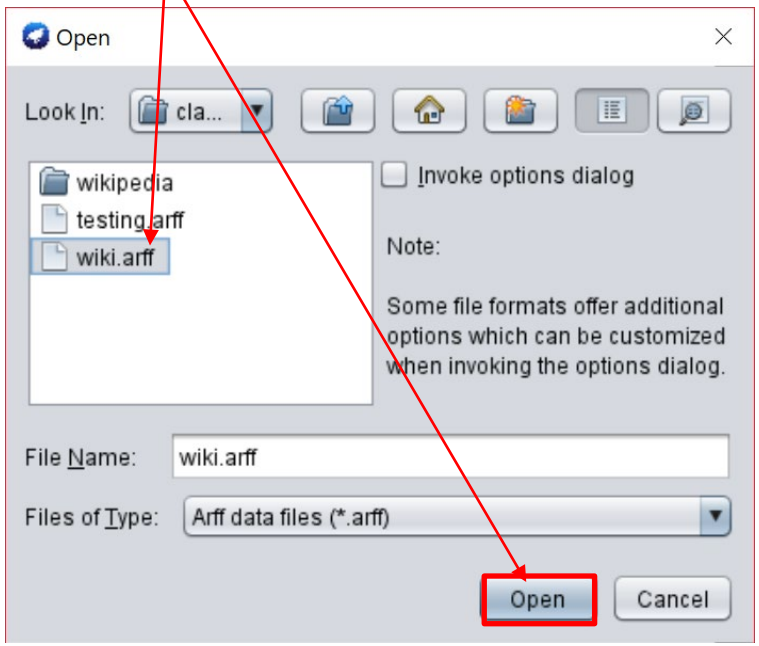


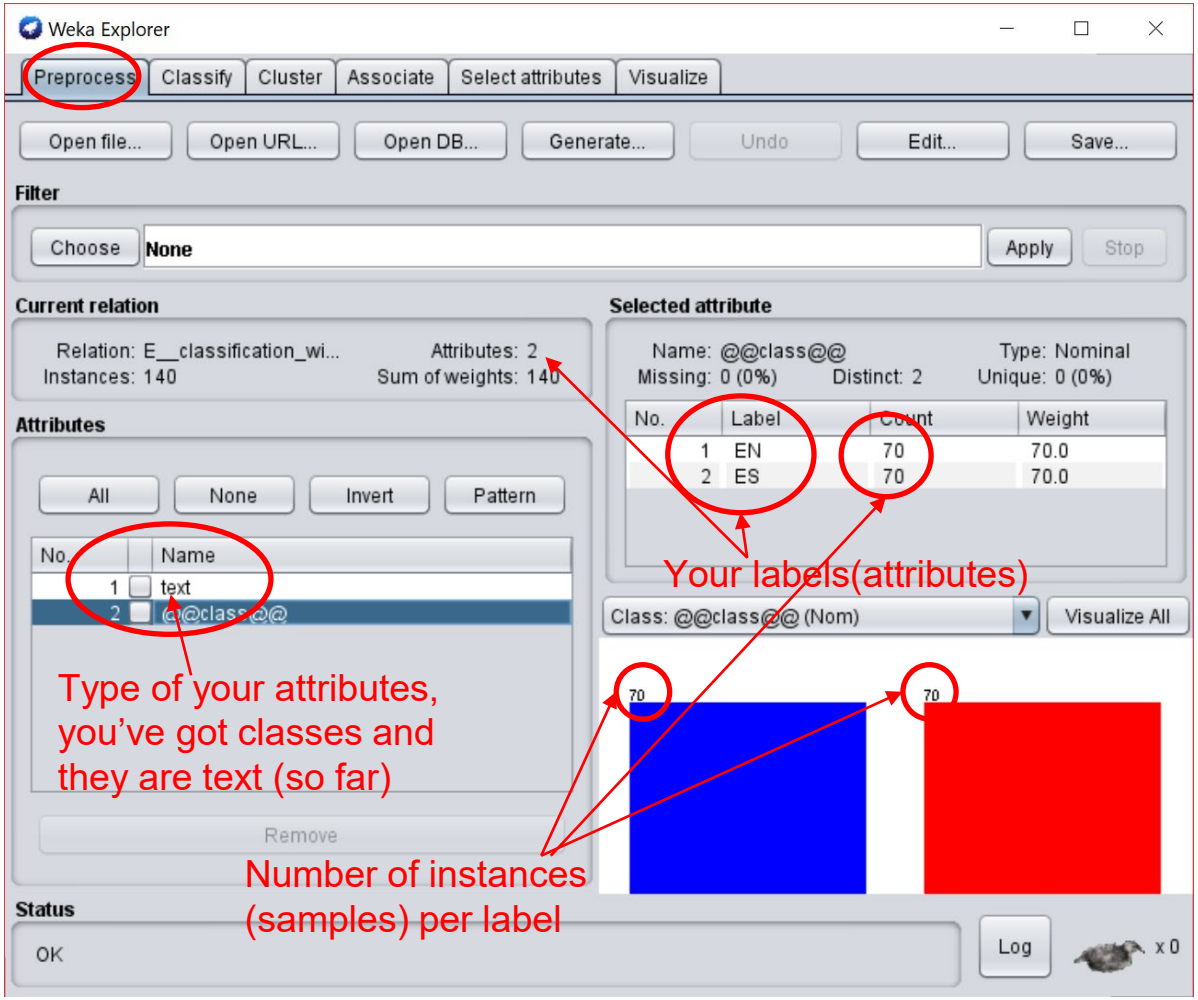
# PREPARE DATASET AND EXTRACT FEATURES

- We start by loading the arff we created into WEKA Explorer to view the contents.
- Navigate to your arff file and click Open



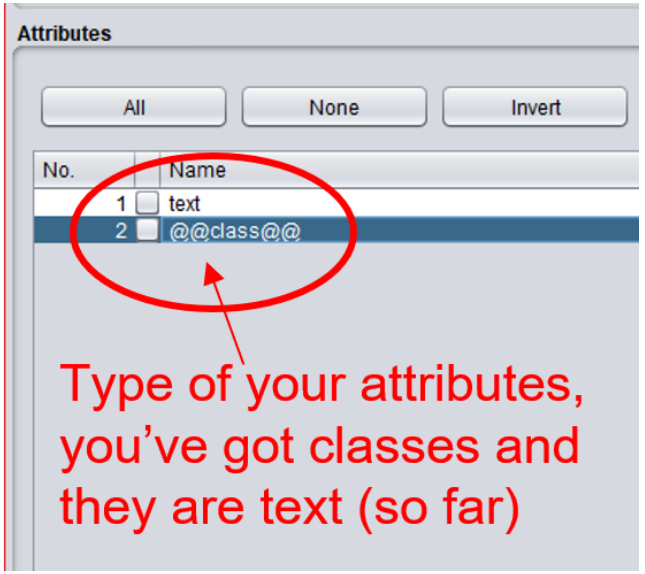
# PREPARE DATASET AND EXTRACT FEATURES



# PREPARE DATASET AND EXTRACT FEATURES

As shown when loading the arff the data is made of “text”.
Algorithms deal with numbers not textual data.
Therefore we need to convert our text (string) attributes into set of numbers.

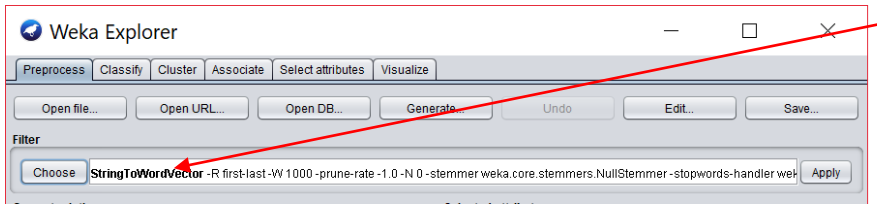
The numbers represent word occurrence information from the text.

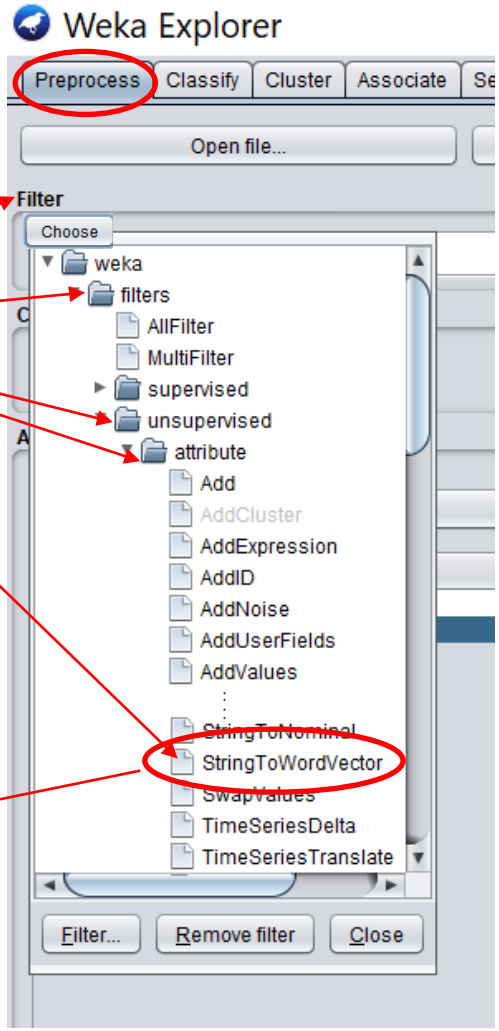


# PREPARE DATASET AND EXTRACT FEATURES

To convert the textual data into numeric attributes we will use a WEKA method (filter) called “StringToWordVector”. The method is “unsupervised” which means it extract word occurrence information without needing an input from you.

Select and click on “StringToWordVectors” to see more properties

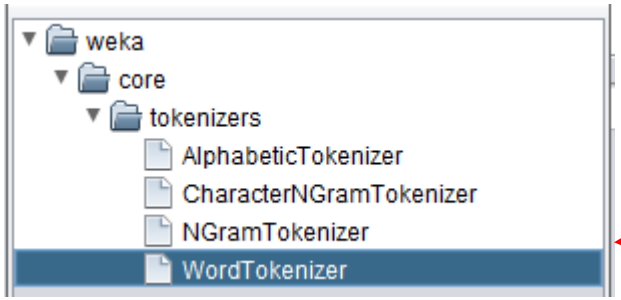


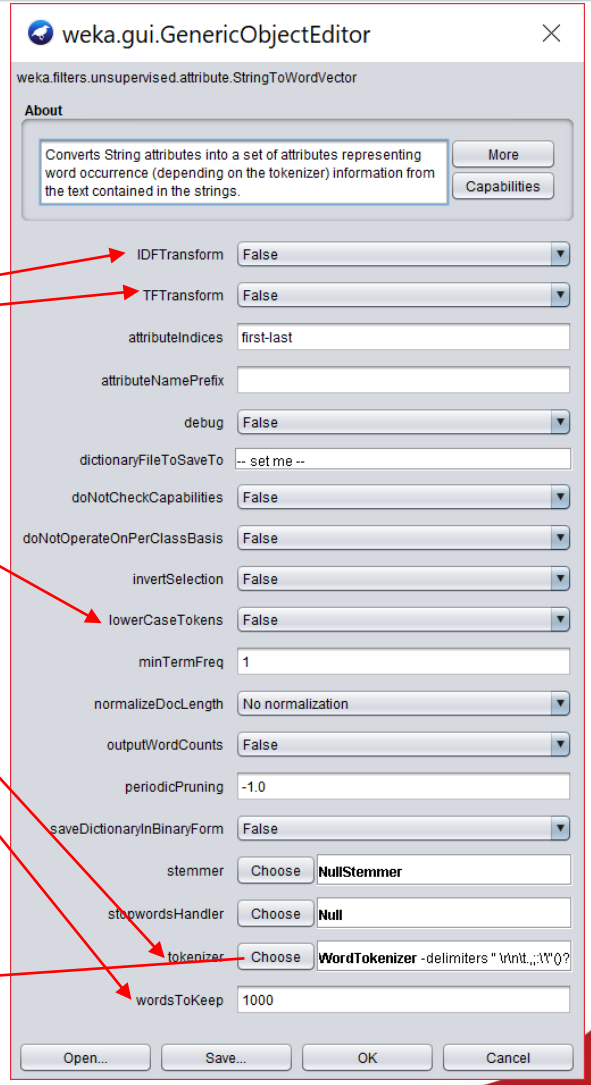



# PREPARE DATASET AND EXTRACT FEATURES

StringToWordVector extract features from the text mainly related to frequencies.
The options on the right helps customise what features to extract.

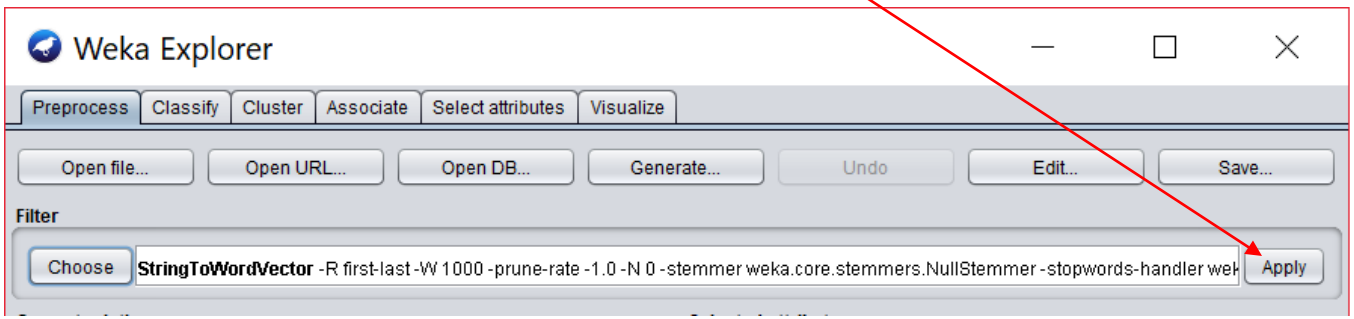
For example you can control how to split words



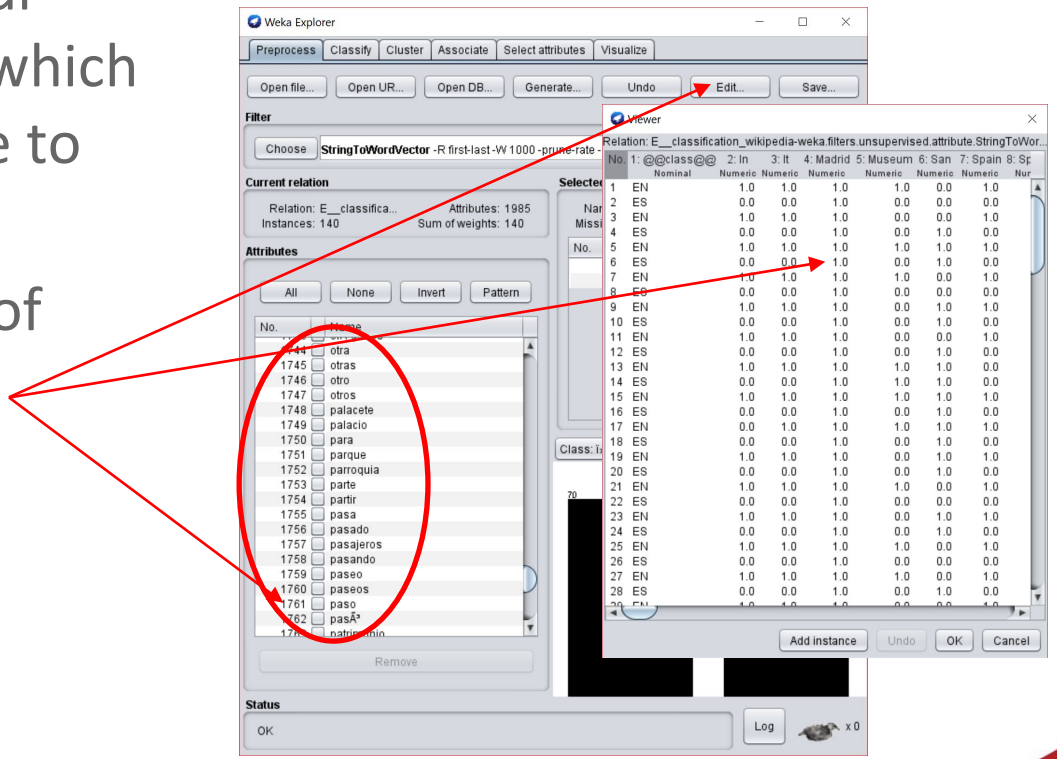
# PREPARE DATASET AND EXTRACT FEATURES

For now leave the default options and click on apply to convert the textual data into numbers



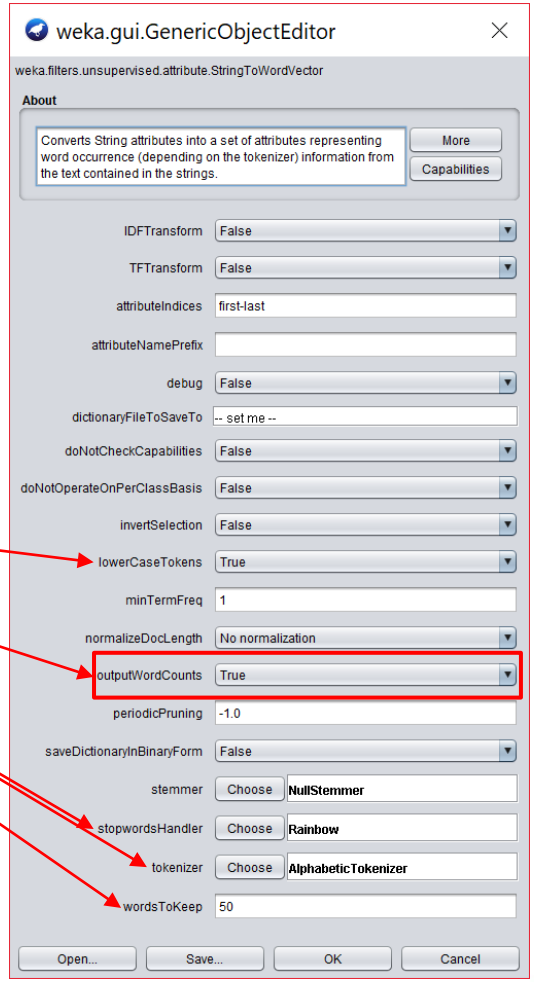
# PREPARE DATASET AND EXTRACT FEATURES

- As soon as you click apply you'll notice an ordered list of numbers and terms appearing instead of the "text" attribute.
  - Those are now your numeric features which algorithms can use to train a classifier.
  - Note the number of features extracted



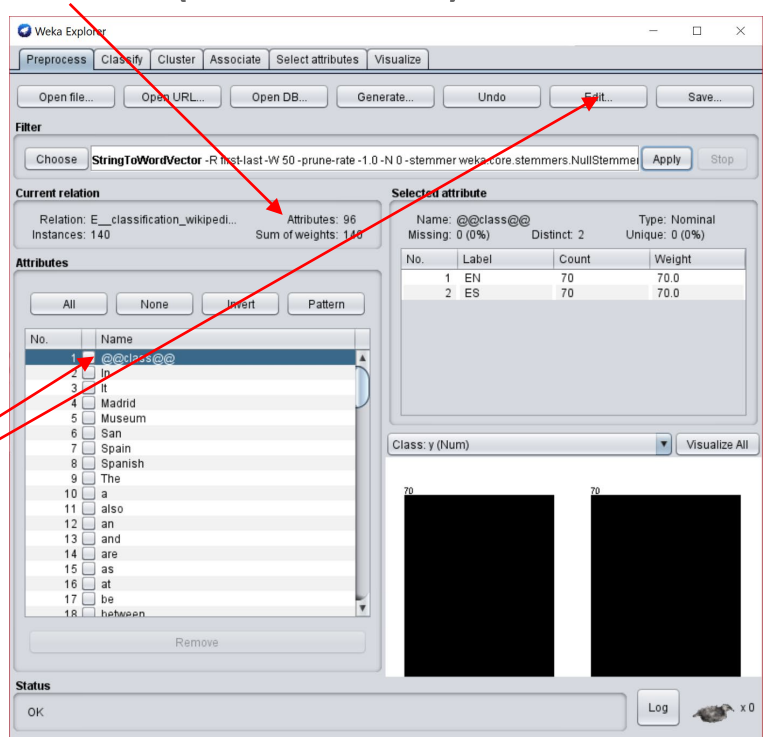
# PREPARE DATASET AND EXTRACT FEATURES

- Let's try reducing number of features.
  - Click "Undo" to cancel the previous operation.
  - Click on StringToWordVector and set the following.
  - Click OK
  - Then Click Apply.



# PREPARE DATASET AND EXTRACT FEATURES

- We now have “approx.” 50-100 words (attributes).
  - Notice the class @@class@@ is now the first in the list.
  - We need to push it back to the end as it’s not a feature.
  - Click on “Edit…”

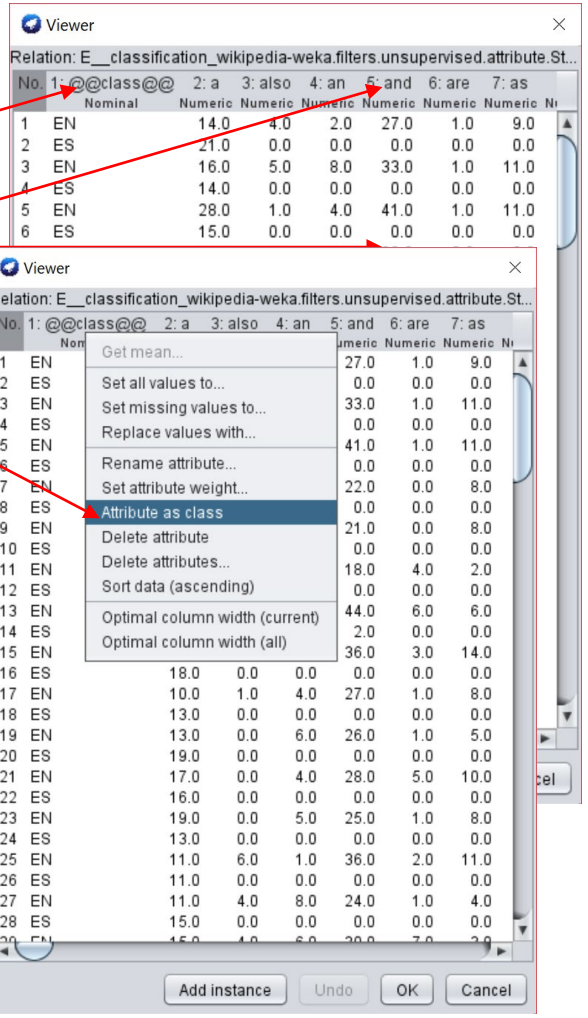


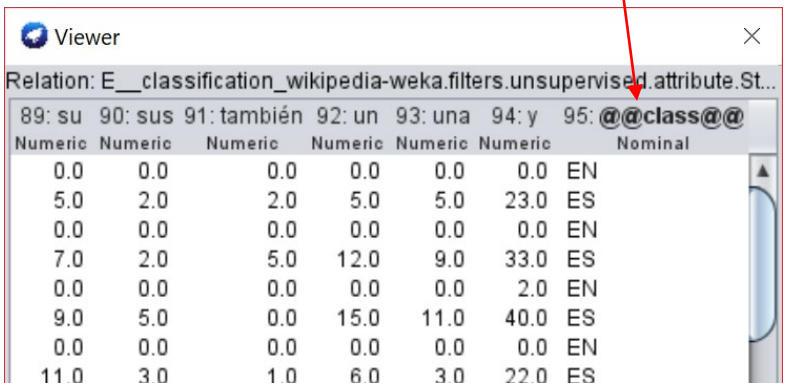
# PREPARE DATASET AND EXTRACT FEATURES

- You'll notice:

1. The @@class@@ is the first attribute.
2. Word count for each of the ~50 words.

- Right click the @@class@@ column and click "Attribute as Class", you'll notice it's the last attribute now.



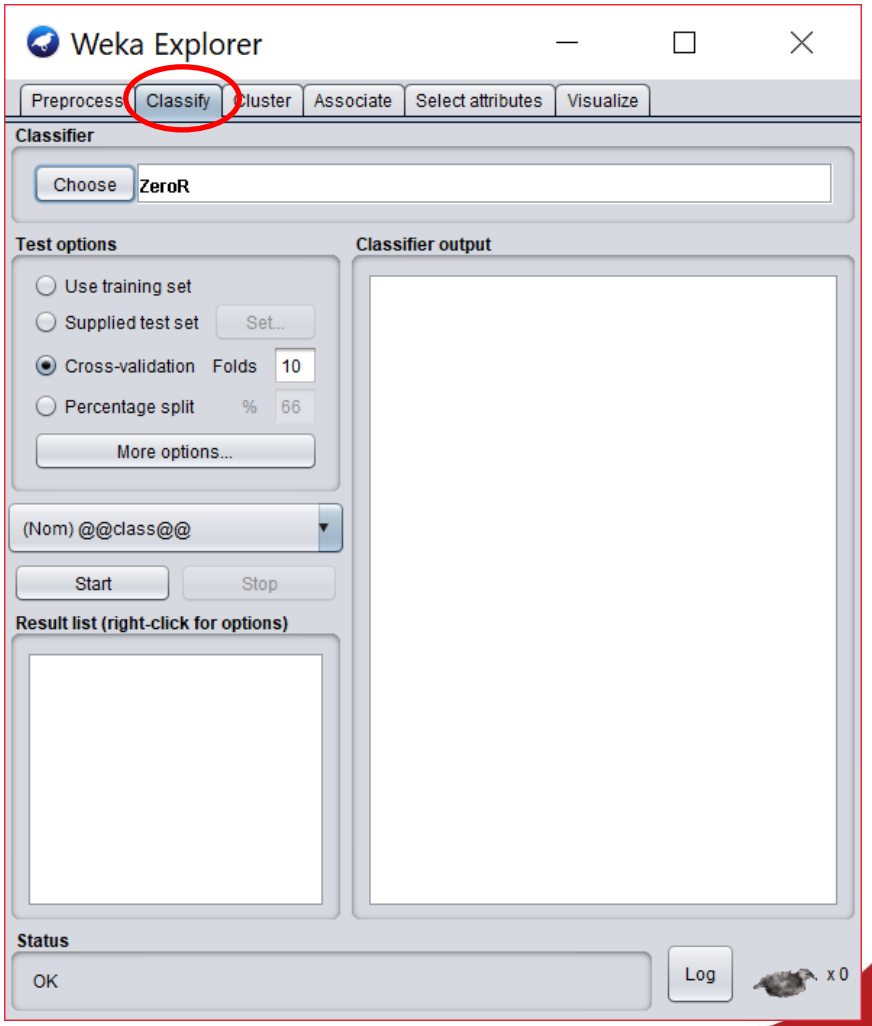



# Steps

1. Convert dataset into ARFF file format.
2. Prepare dataset and extract features
3. **Train a classifier using training data**
4. Create a classification model
5. Test the classification model

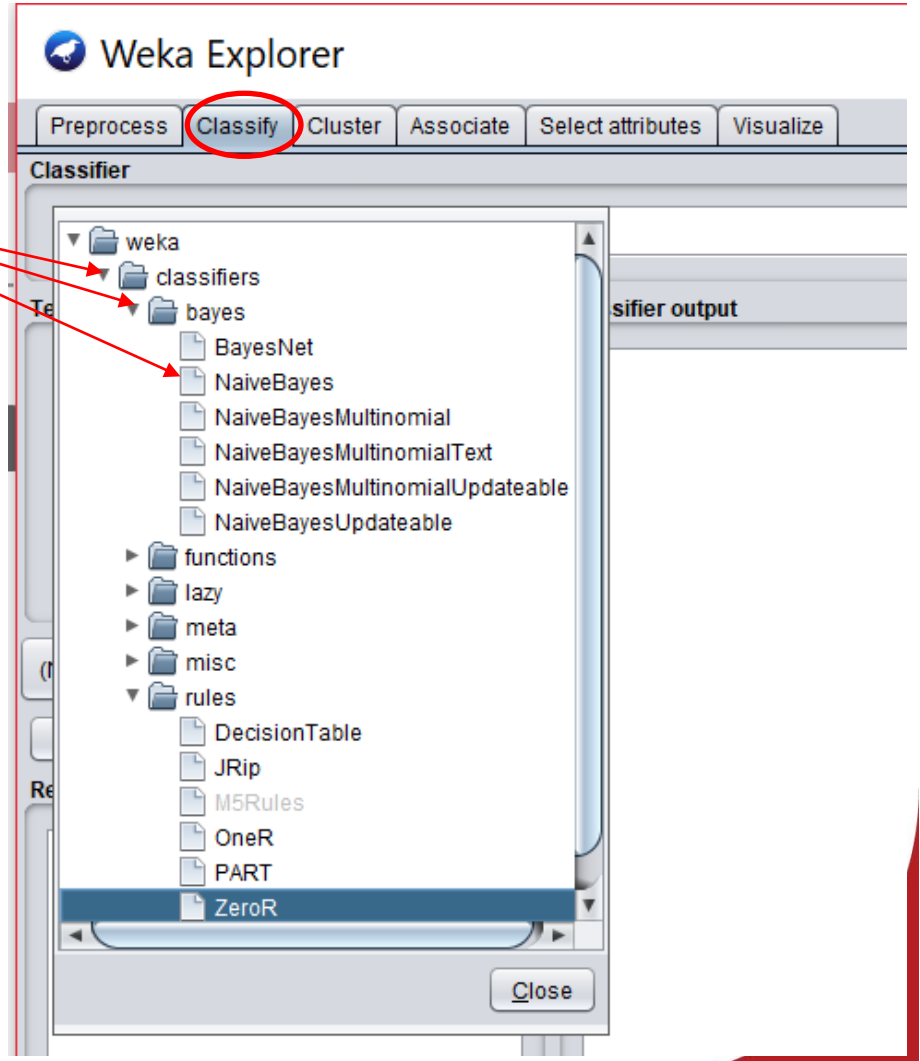
# TRAIN A CLASSIFIER USING TRAINING DATA

- Now that we have extracted the features from our dataset we can now start to train a classifier.
- Start by selecting the "Classify" tab in the WEKA Explorer window.



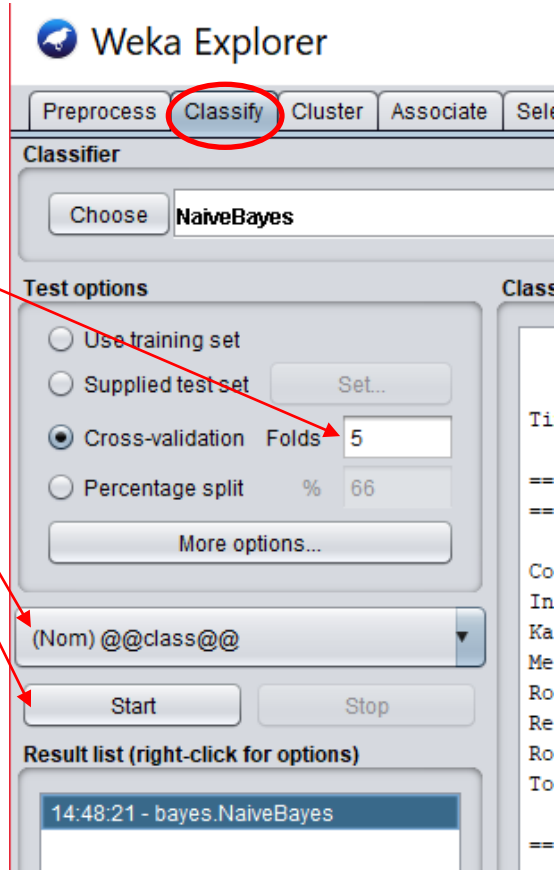
# TRAIN A CLASSIFIER USING TRAINING DATA

- Choose a classifier to begin with. Let's start with a Naïve Bayes Classifier
- Click to select the classifier.



# TRAIN A CLASSIFIER USING TRAINING DATA

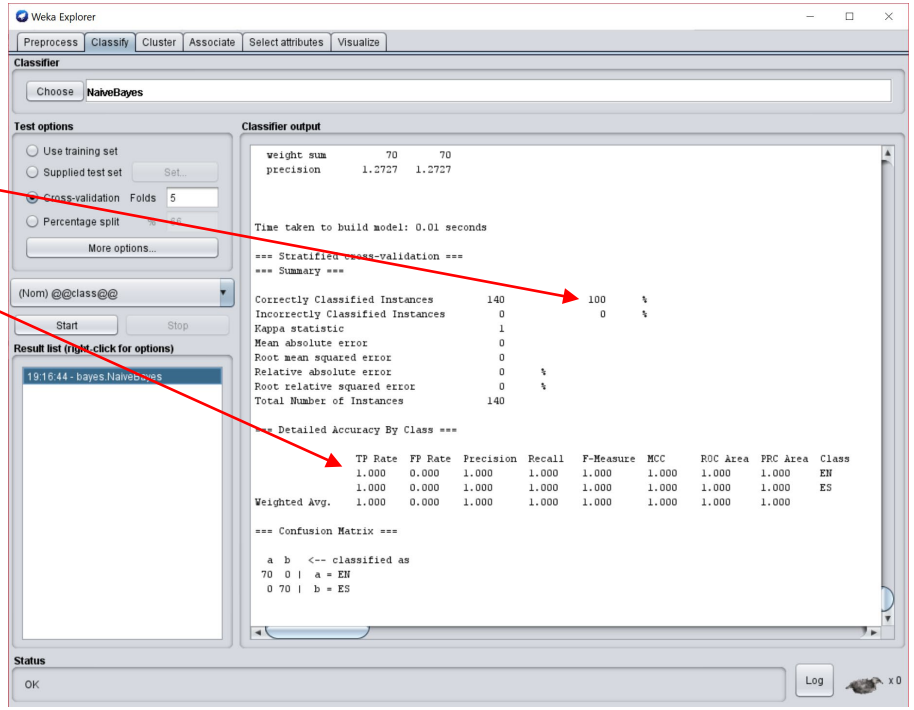
- Select the number of folds (training k-folds). Select 5 and make sure the (Nom) @@class@@ is selected (ES and EN)
- Then click on Start to start the training process.



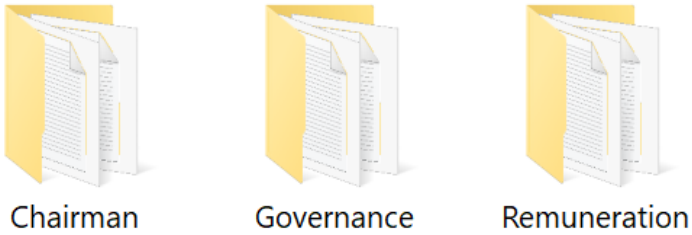
# TRAIN A CLASSIFIER USING TRAINING DATA

- You'll see a summary of the training process using Naïve Bayes such as:
- Classifier's accuracy
- And a confusion matrix

- This is basically saying that training process expects a 100% accuracy when classifying new unseen samples.



# Harder task




Classifying UK financial sections (3 classes)

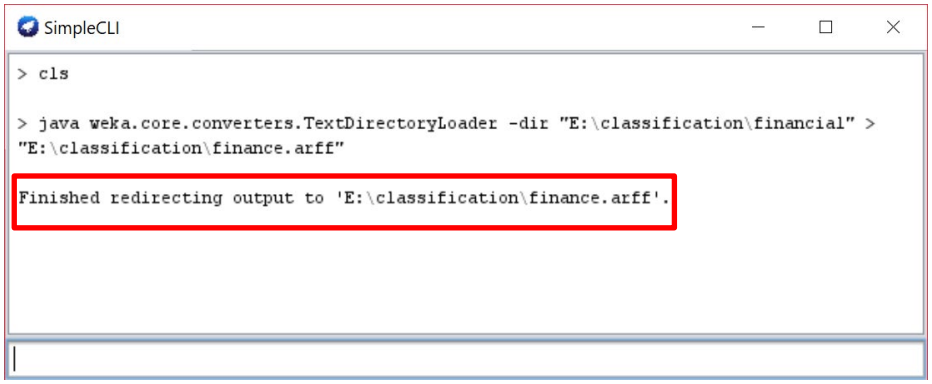
# Repeat Steps

1. Convert dataset into ARFF file format.
2. Prepare dataset and extract features
3. Train a classifier using training data

# REPEAT STEPS

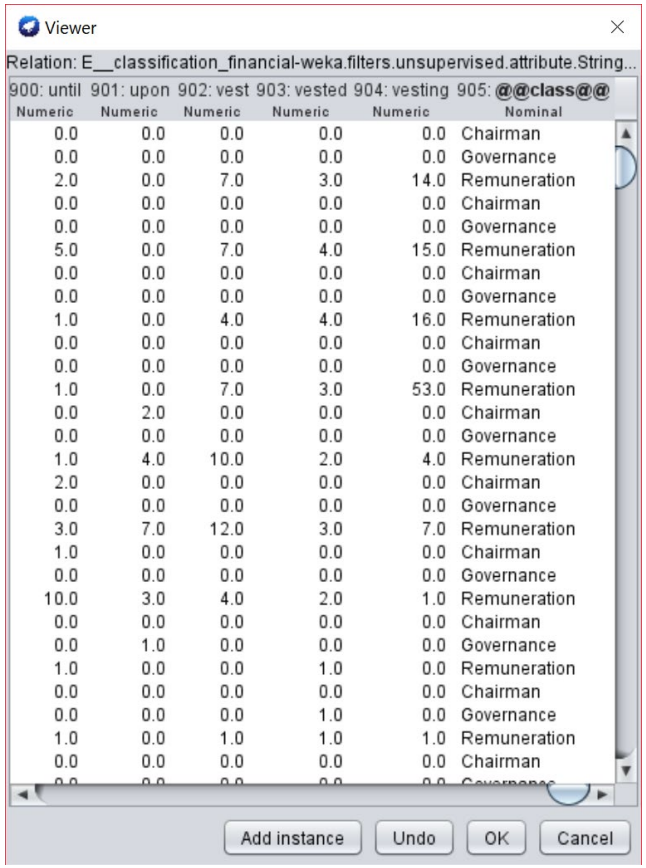
1. Convert dataset into ARFF file format.

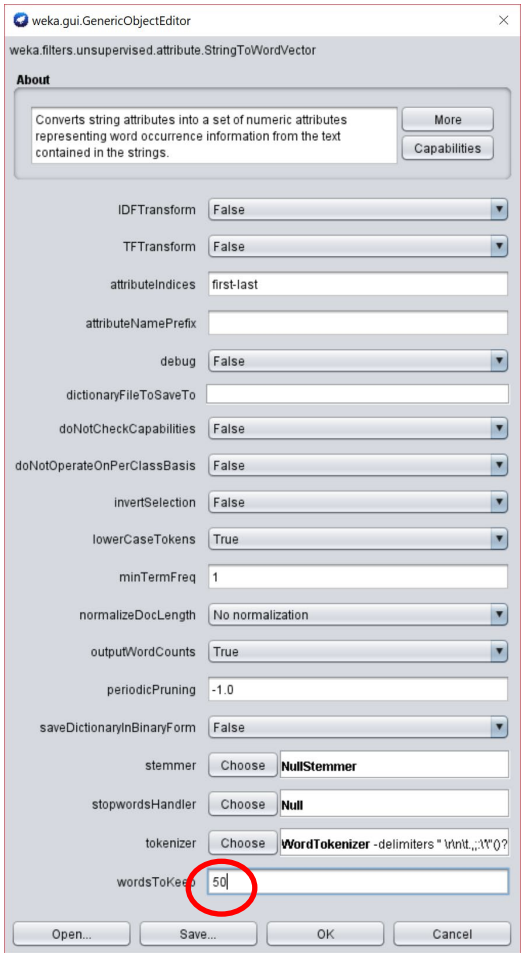
java weka.core.converters.TextDirectoryLoader -dir "E:\classification\financial" > "E:\classification\finance.arff"




# REPEAT STEPS

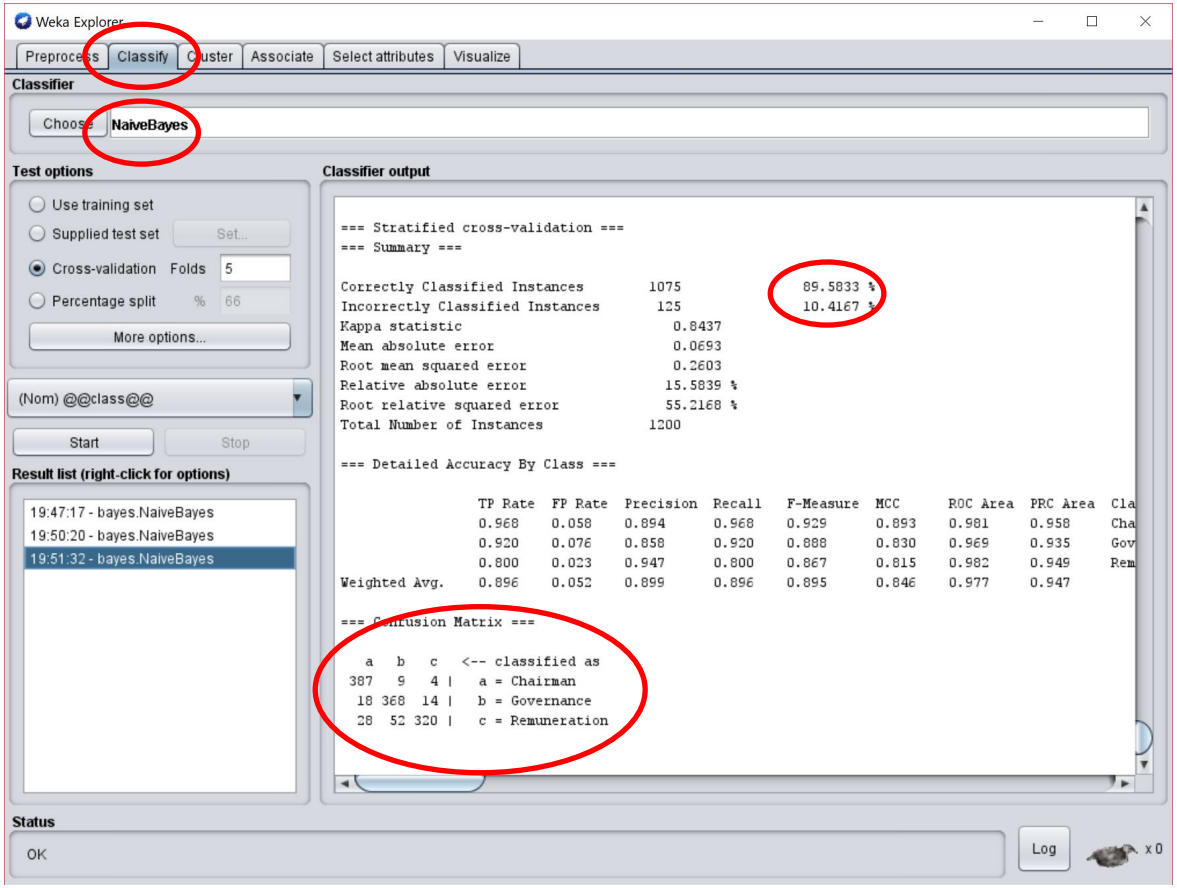
2. Prepare dataset and extract features



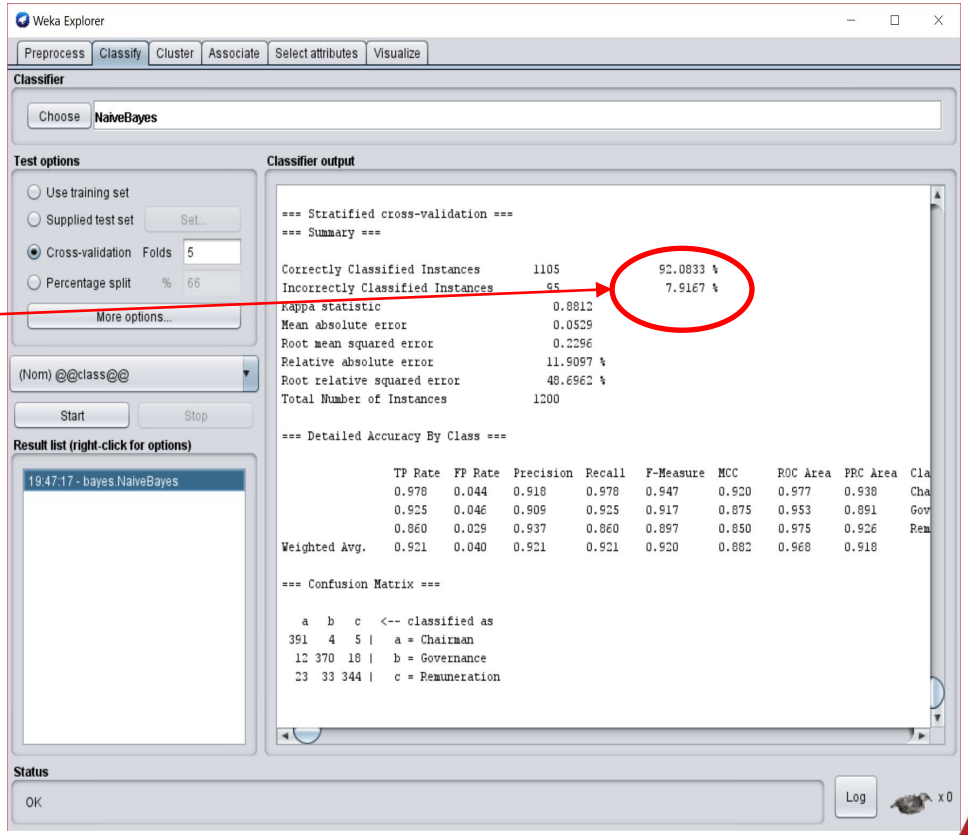
# REPEAT STEPS

## 3. Train a classifier using training data



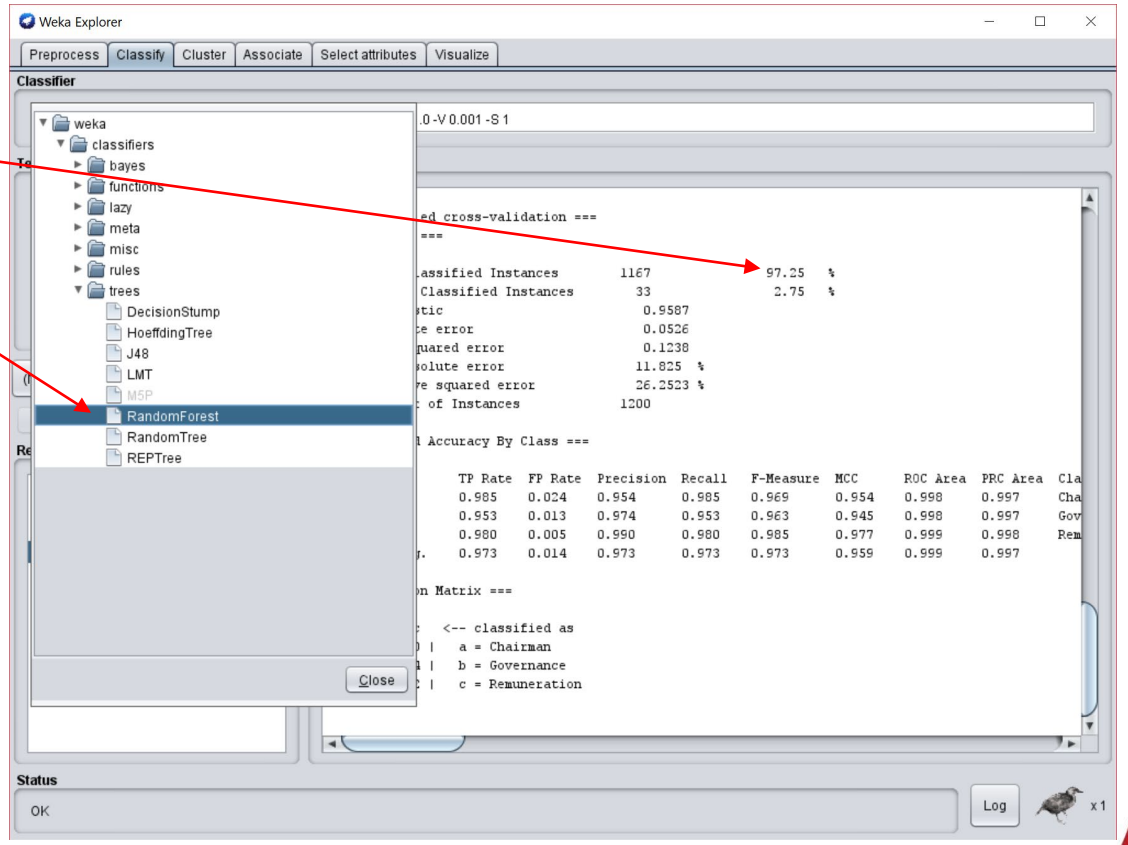
# TRAIN A CLASSIFIER USING TRAINING DATA

- Can we increase accuracy?
- Let's go back to the Preprocess tab, undo until we are back to the beginning and then ask StringToWordVector to keep 500 words instead.
- Apply, then go back to Classify and run the classifier again.
- This time I got 92.08% accuracy.



# TRAIN A CLASSIFIER USING TRAINING DATA

- Let's try Random Forest
- Basically, you'll have to experiment with few classifiers and different features to arrive at your best model and still avoid overfitting.



# Steps

1. Convert dataset into ARFF file format.
2. Prepare dataset and extract features
3. Train a classifier using training data
4. **Create a classification model**
5. Test the classification model

# CREATE A CLASSIFICATION MODEL

- In order to measure the real accuracy of your classifier you need to test it on unseen examples (those are examples that were not included in the training process and that the algorithm doesn't have a prior knowledge about (**?**)).

- To be able to do so we need to create a classification model.

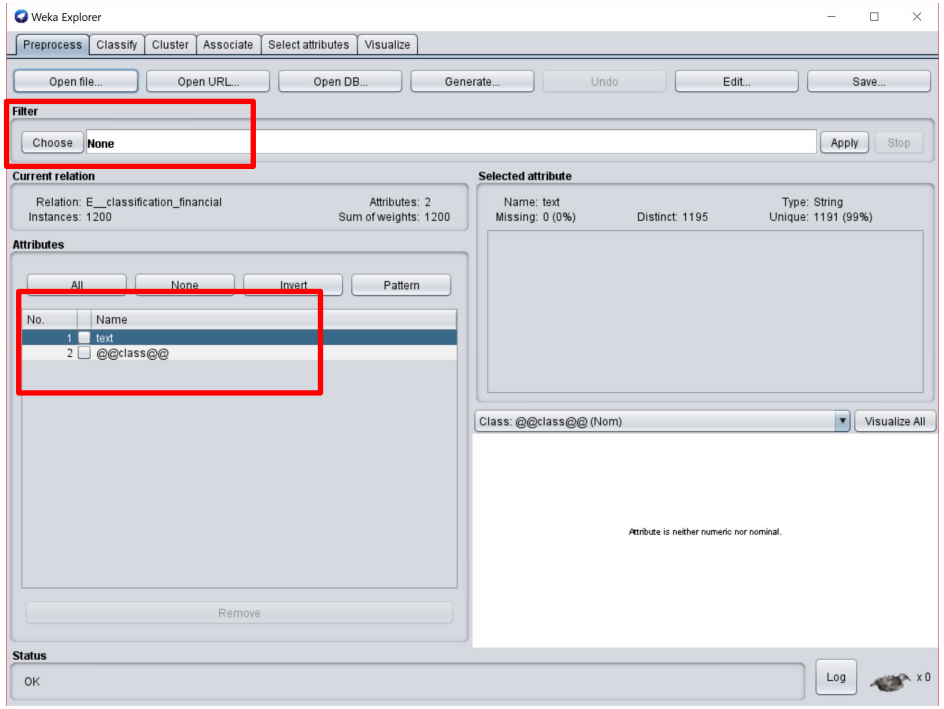
# CREATE A CLASSIFICATION MODEL

- WEKA requires that you apply the same filter to training and testing datasets when extracting features.
- As mentioned before, you must use **identical features** for both training and testing as otherwise the algorithms may not work.

- To do this we need to create a "filtered classifier". That is a classifier combined with a filter to extract the same features from both the training and testing datasets.

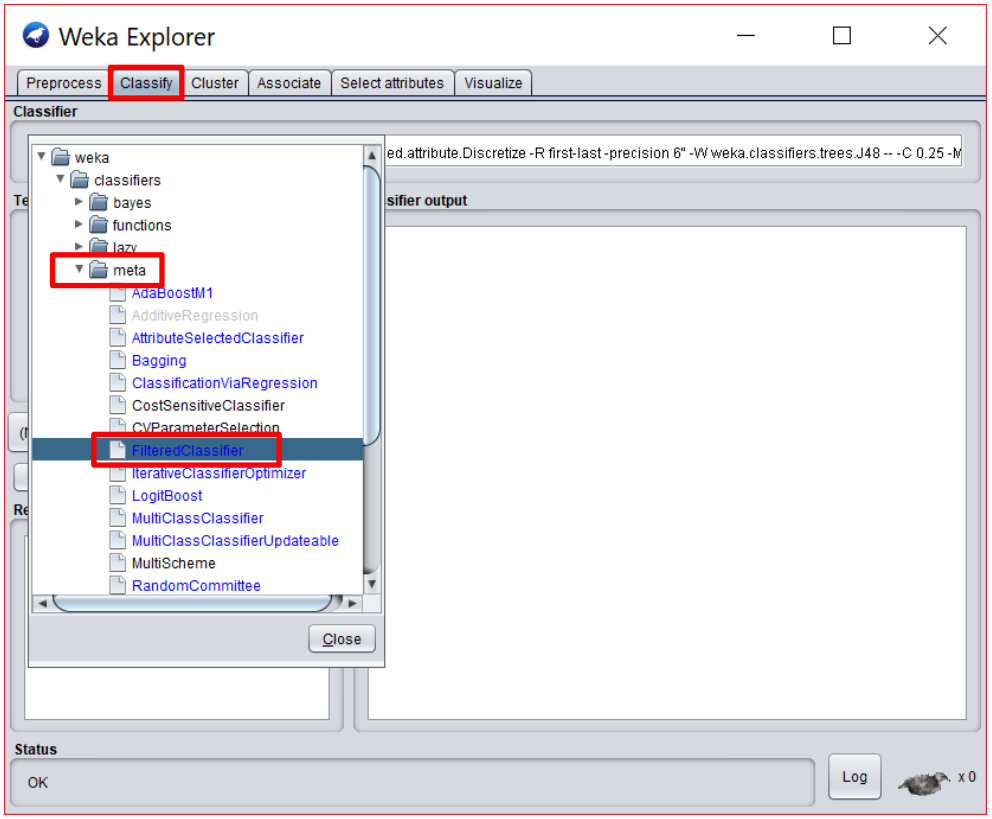
# CREATE A CLASSIFICATION MODEL

- Start by loading your training arff file (**finance.arff**).
- This time **do not apply** the StringToWordVector filter, yet!



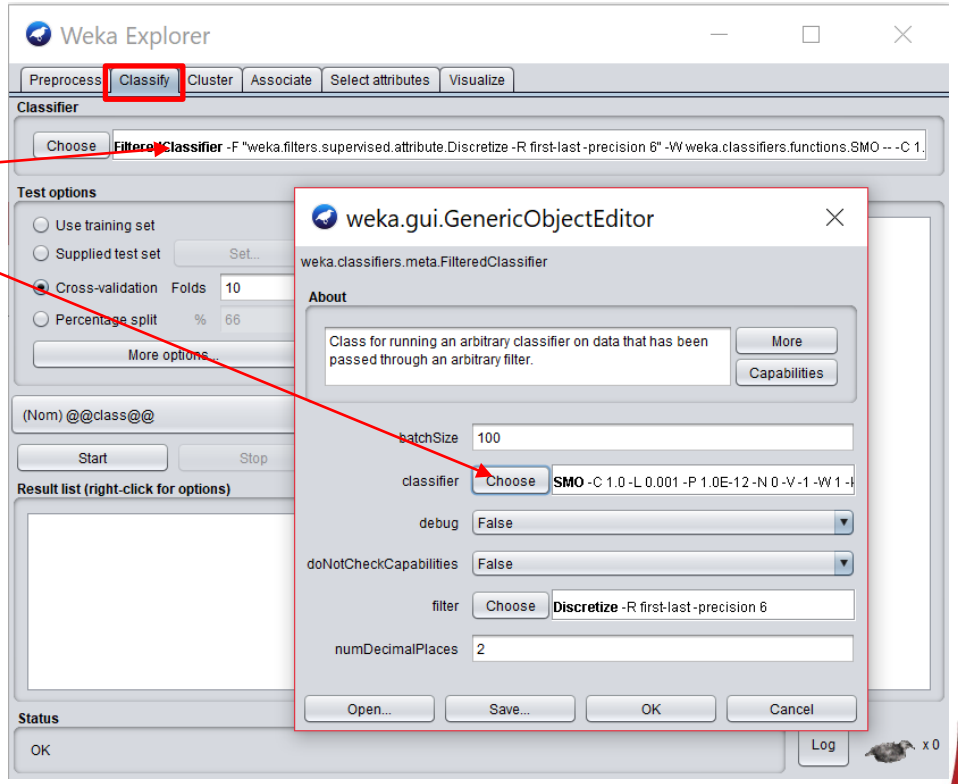
# CREATE A CLASSIFICATION MODEL

- Move on to the Classify Tab and choose the “Filtered Classifier” from the Classifier Chooser menu.



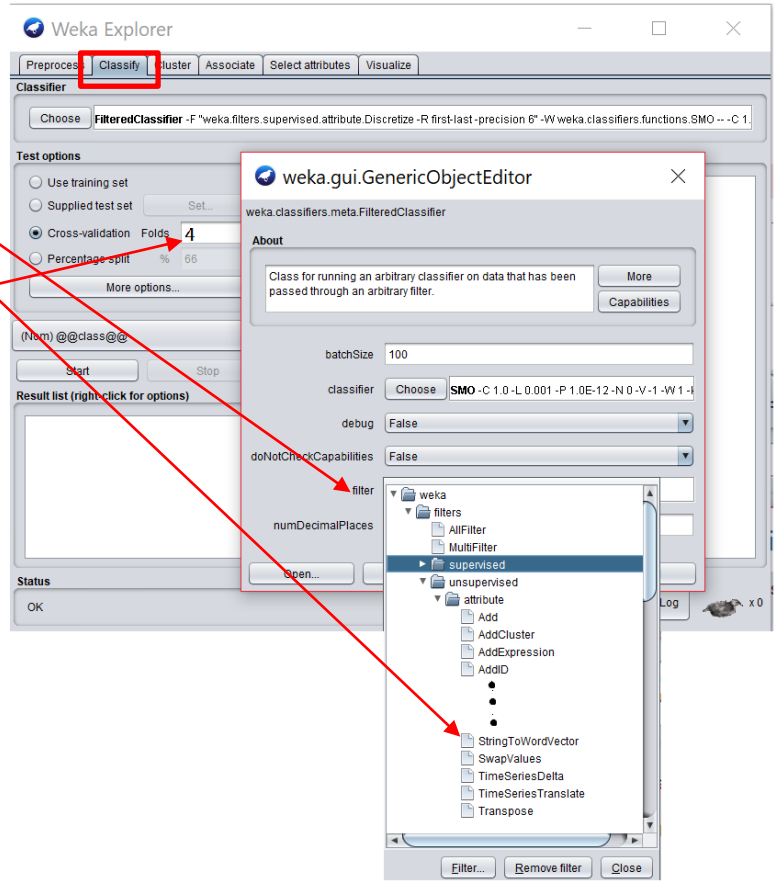
# CREATE A CLASSIFICATION MODEL

- Click on FilteredClassifier and Choose SMO from the functions classifiers.
- SMO is the SVM classifier in WEKA.



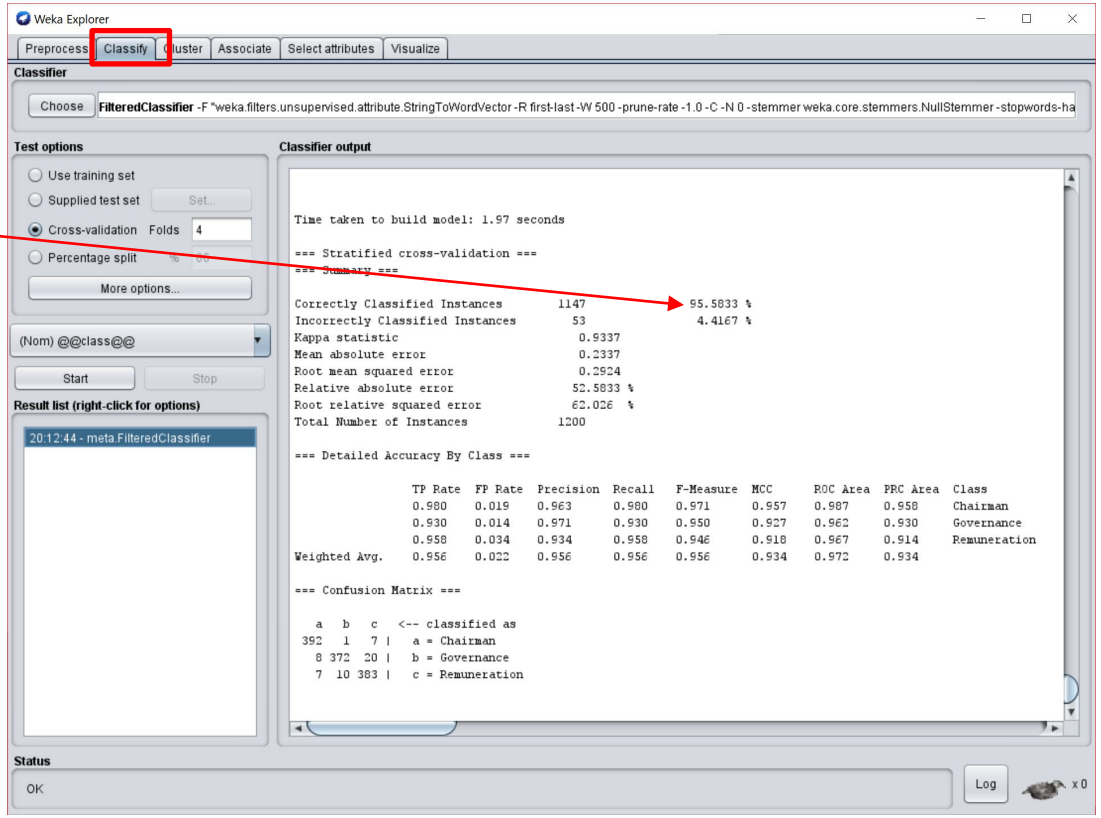
# CREATE A CLASSIFICATION MODEL

- Click on filter to choose StringToWordVector (keeping 500 alphabetical words)
- Click Ok
- Choose 4 Cross Validation Folds
- Click Start



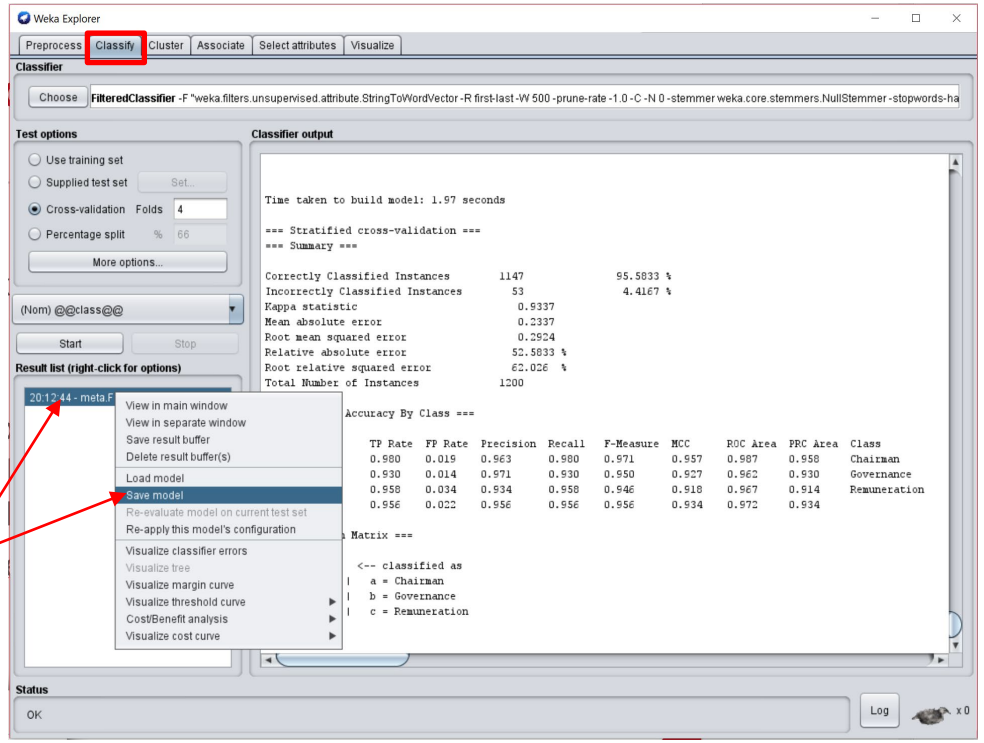
# CREATE A CLASSIFICATION MODEL

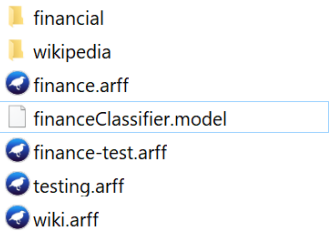
- Classification Results will show in the Classifier output window



# CREATE A CLASSIFICATION MODEL

- Right click the SMO Filtered classifier and click on Save model
- Save the model to your machine next to the arff file so you can find it later on.
- Call it aaerClassifier and click Save.



# Steps

1. Convert dataset into ARFF file format.
2. Prepare dataset and extract features
3. Train a classifier using training data
4. Create a classification model
5. **Test the classification model**

# FINANCIAL TESTING DATA

- To save your time I created a testing file for you

finance-test.arff

1-5: chairman statements
6-10: governance statements
11-15: remuneration statements

```
@relation E__classification_financial

@attribute text string
@attribute @@class@@ {Chairman,Governance,Remuneration}

@data

'Euan Worthington Paul LoudonChairman Chief Executive Officer5',?
'outlook with optimism.Otto SchmidChairman 10 October 2002',?
'Financial statements Other information Overview Strategic report',?
```

# TEST THE CLASSIFICATION MODEL

- Now that we have a SVM Classifier (Model) we want to test it on unseen data.
- We'll use the **finance-test.arff** file.
- The classifier doesn't know the output and only you know so you can judge the classifier's real accuracy.

# FINANCIAL TESTING DATA

- To save your time I created a testing file for you

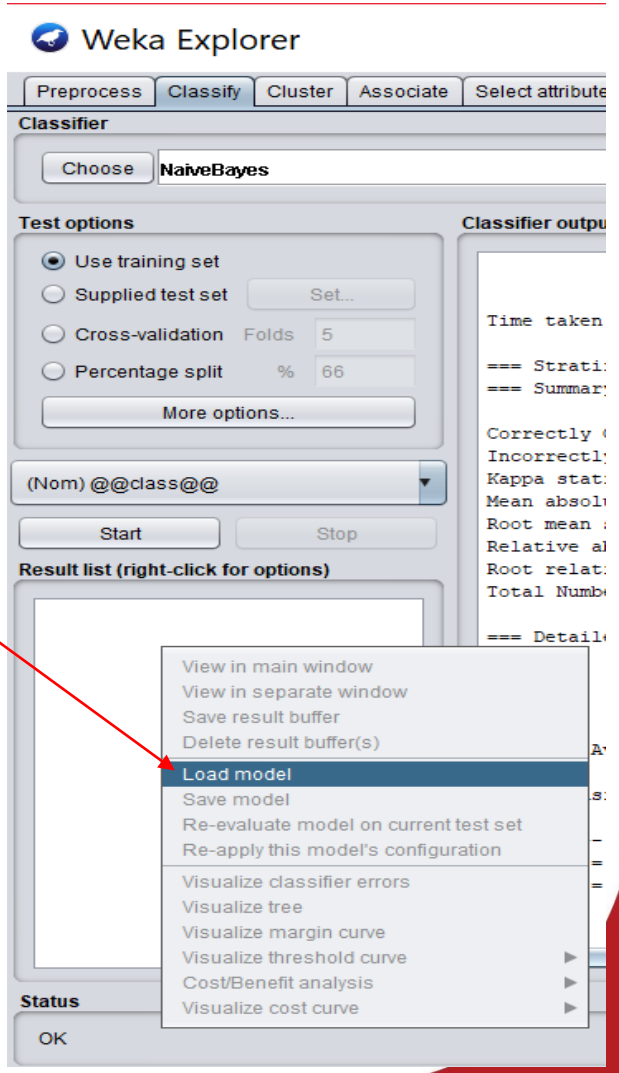
finance-test.arff

1-5: chairman statements
6-10: governance statements
11-15: remuneration statements

```
@relation E__classification_financial

@attribute text string
@attribute @@class@@ {Chairman,Governance,Remuneration}

@data

'Euan Worthington Paul LoudonChairman Chief Executive Officer5',?
'outlook with optimism.Otto SchmidChairman 10 October 2002',?
'Financial statements Other information Overview Strategic report',?
```

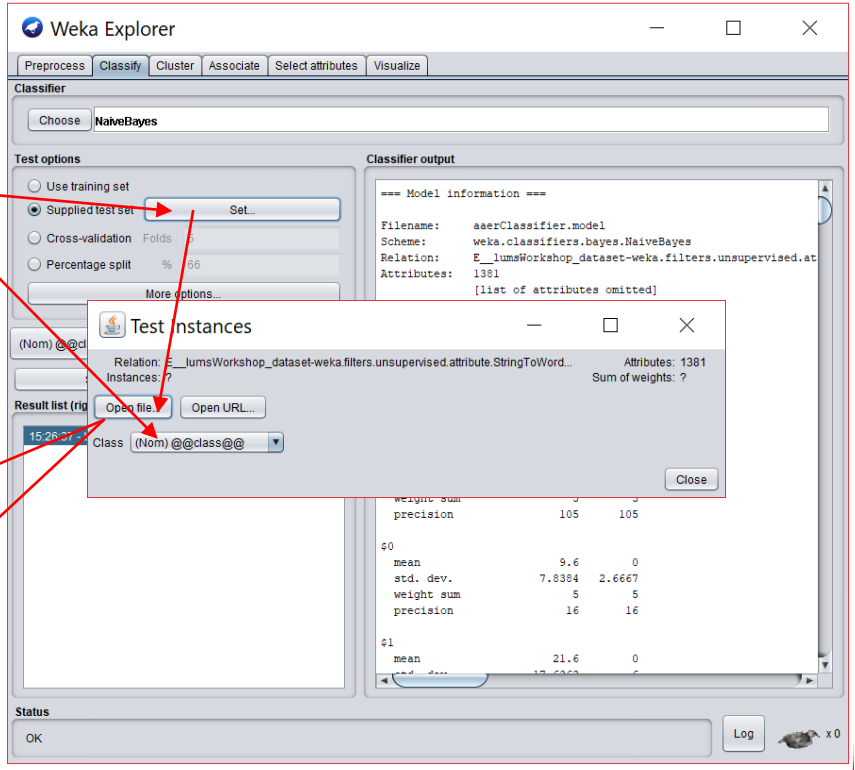
# TEST THE CLASSIFICATION MODEL

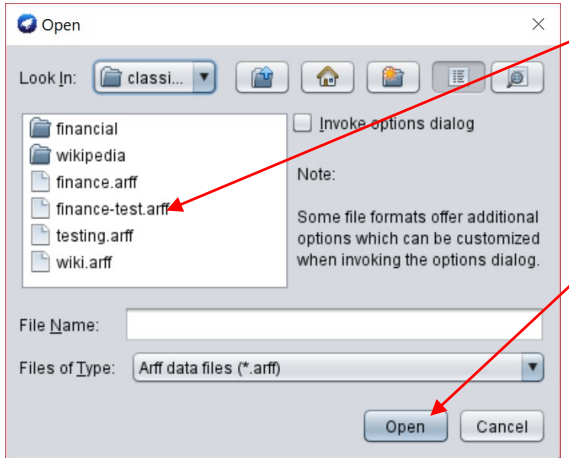
On Weka Classify right click in the result list to load your financeClassifier.model (only if it's not already there).



# TEST THE CLASSIFICATION MODEL

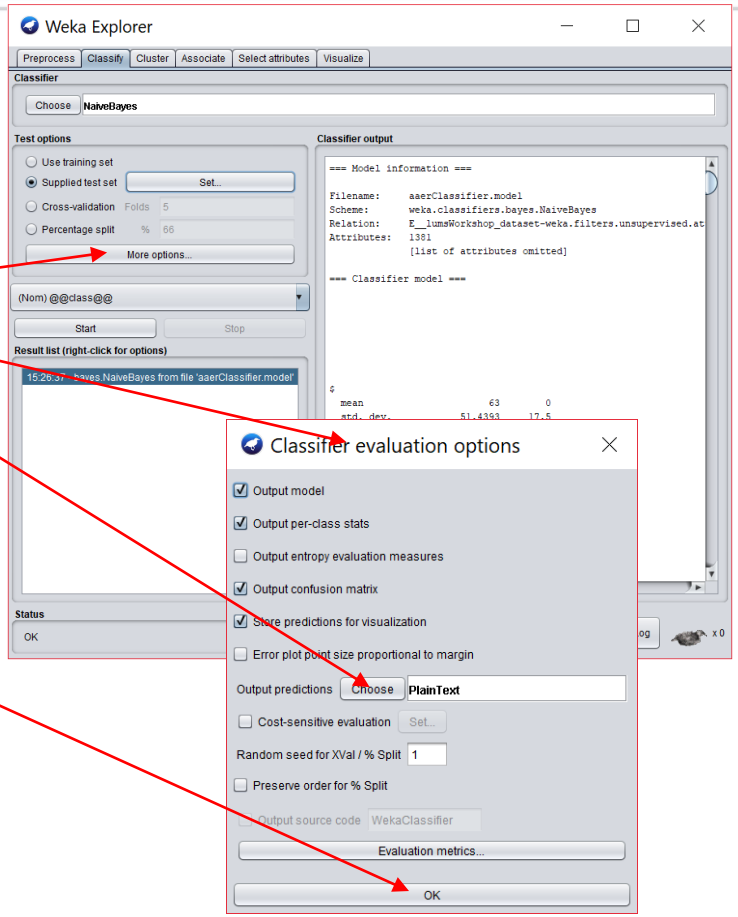
- Now you need to load your testing sample (finance-test.arff).
- Choose Supplied test set
- Make sure (Nom) @@class@@ is selected
- Open your testing arff file and click open then close the dialogue.
- If you get an error message here then your training and testing arff files are incompatible.



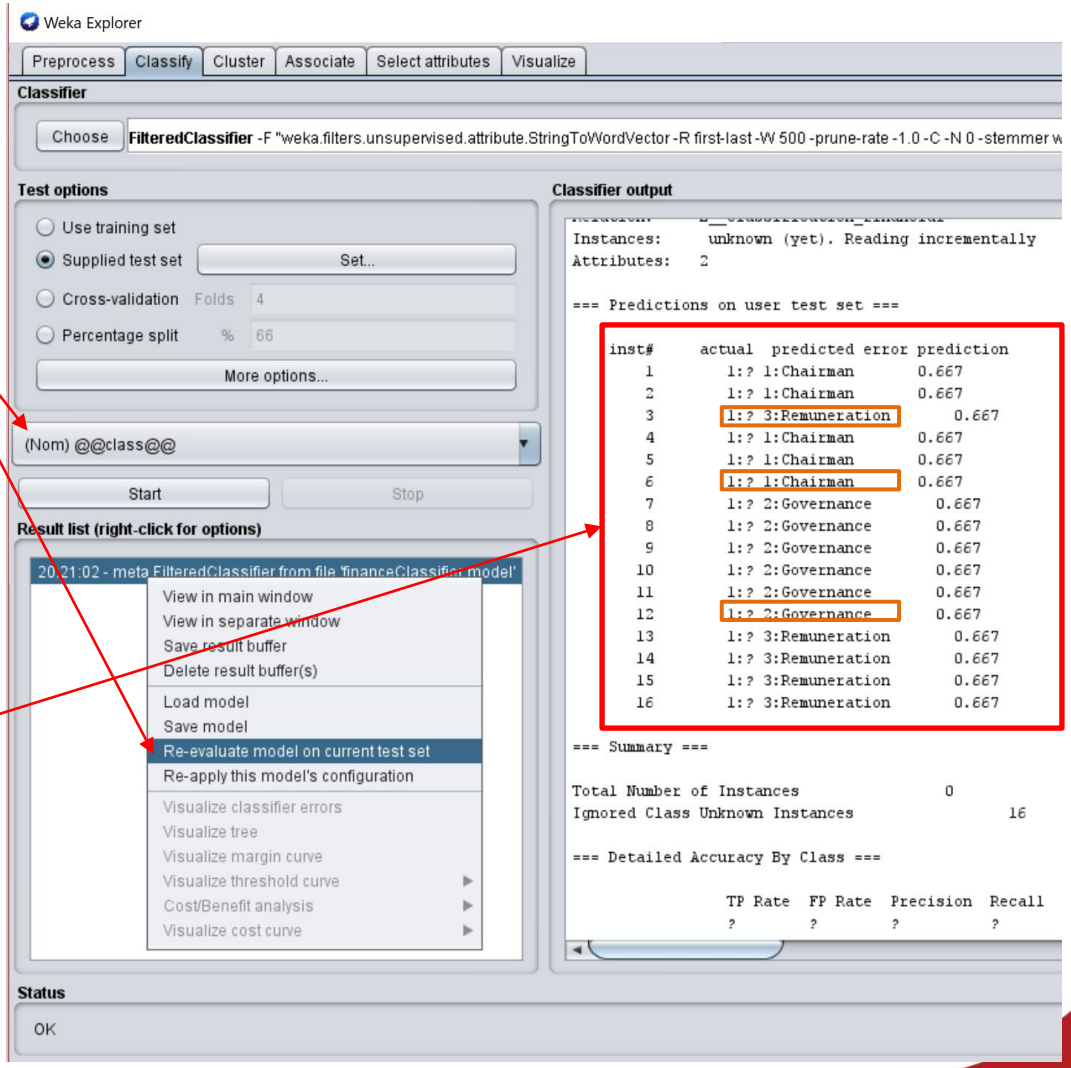
# TEST THE CLASSIFICATION MODEL

- Click on More options..
- Make sure Output predictions is PlainText so you can see the output as Chairman, Governance or Remuneration.
- Then click OK



# TEST THE CLASSIFICATION MODEL

- Make sure (Nom) @@class@@ is selected
- Right Click your model and click Re-evaluate model on current test set.
- You'll notice the predictions in the Classifier output window.



# TRY YOURSELF