Topic - 03
Introduction to Machine Learning: Concept & Fundamentals
-- Classification & Terminologies

☎ +88 01831-661534

🔗 jehadfeni@gmail.com

Classification – based on Generalization

Based on how the Generalization happens, Machine Learning can be divided into:

Instance-based Learning

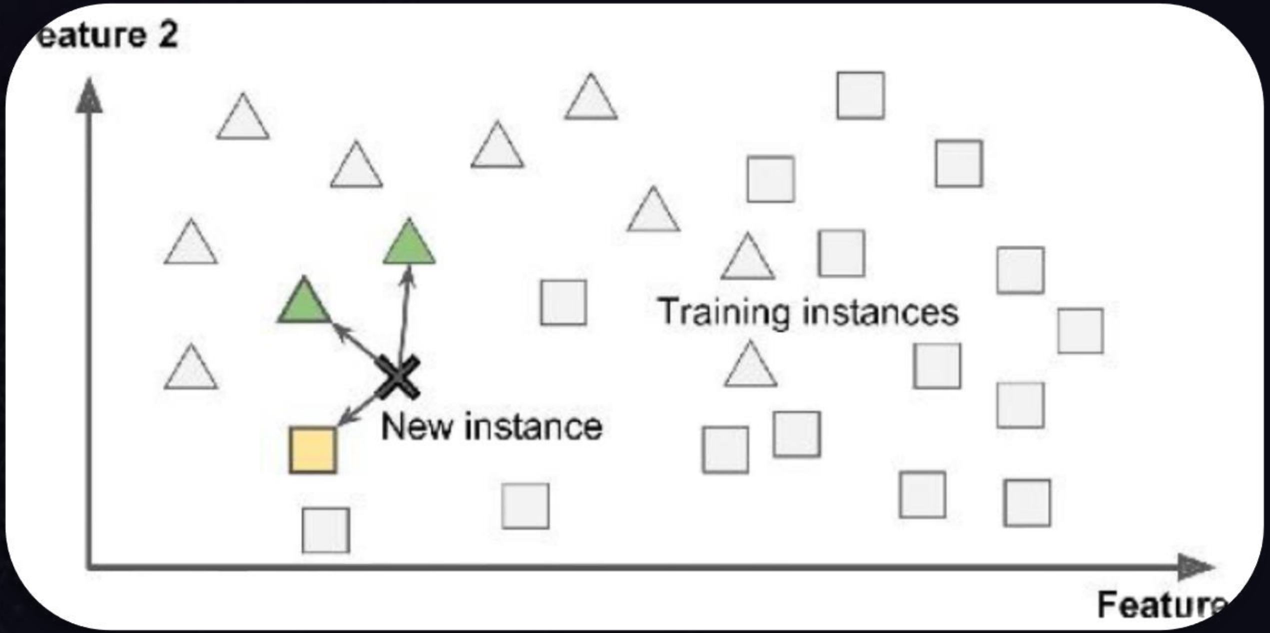
Model-based Learning

Instance-based Learning

In Instance-based learning, the system learns by heart. For example: Let’s say you have built a Spam Filter system which identifies emails that are identical to previously Tagged Spam Emails.

We can update this system by not just only Tagging emails that are identical to Spam emails but also has some similarity with the Spam Email. A simple approach of this can be measuring same words in both the emails. The system will identify email as Spam which already has many words common with a previously known Spam email.

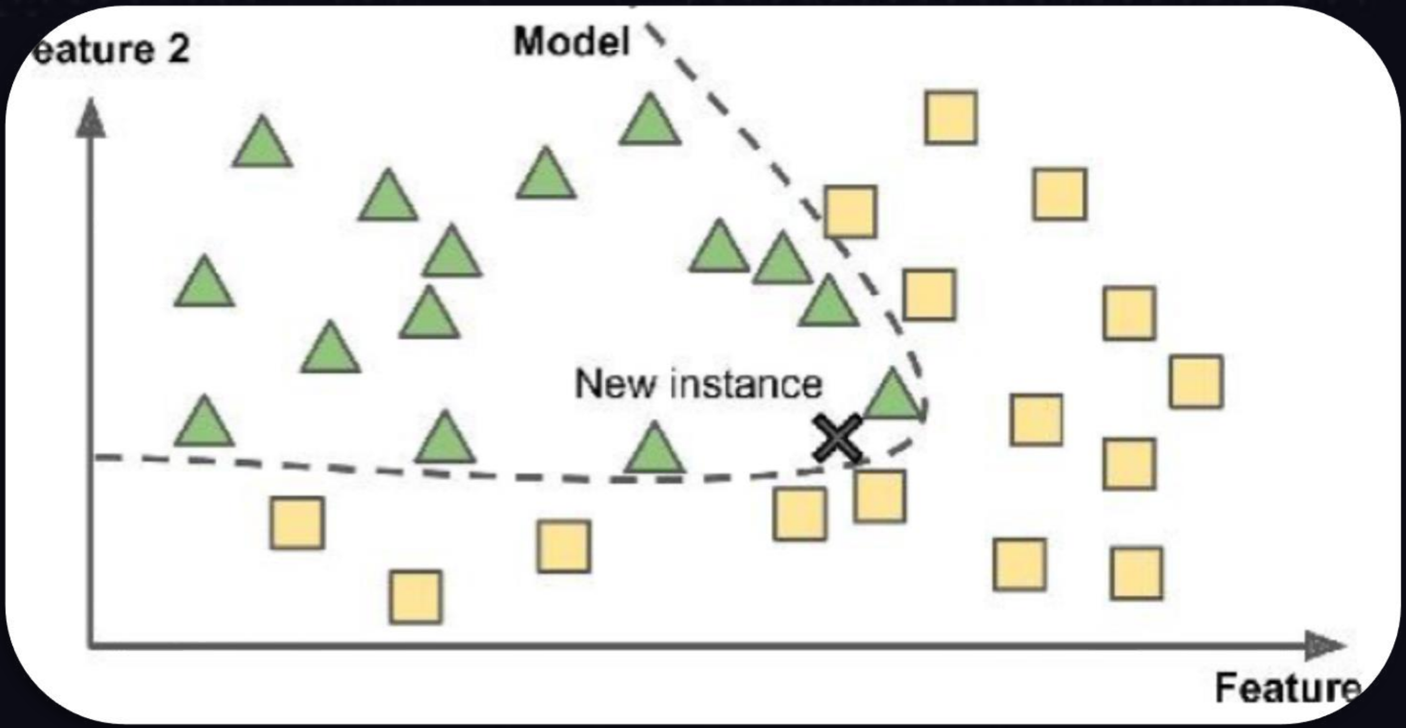
This is called Instance-based learning. The system learns the examples by heart, then generalizes to new cases using a similarity measure.



Model-based Learning

A common way to generalize a machine learning system is based on making a Model from a set of examples. Then save this model somewhere and use this model to predict the Accuracy of the system by giving new data into the system.

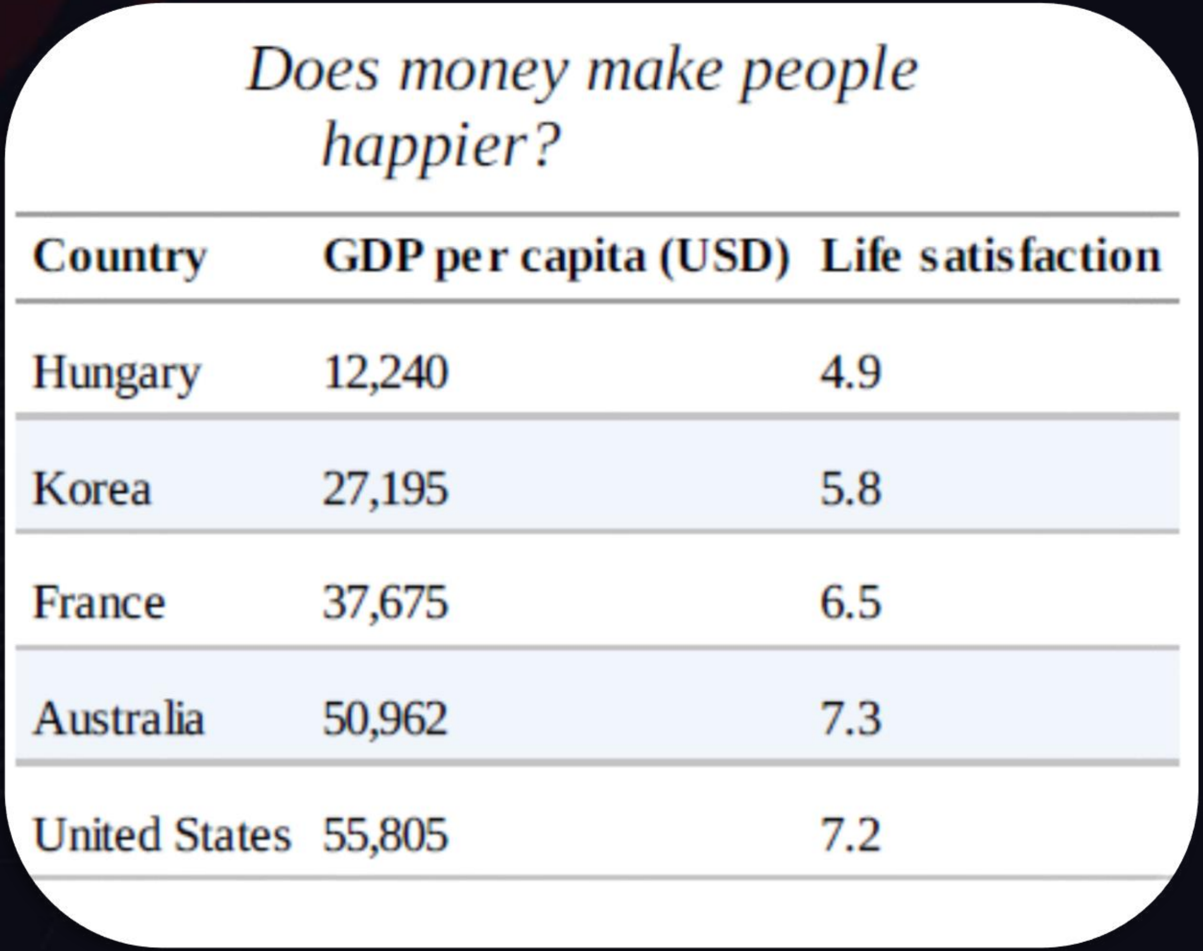
This is called Model-based learning.



Model-based Learning – Example

Suppose you want to know if people get happy with the money! If money makes people happy. So you download the Better Life Index data from the OECD’s website as well as stats about GDP Per Capita from the IMF’s website.

Then you join the tables and sort by GDP per capita.



Does money make people happier?

Country	GDP per capita (USD)	Life satisfaction
Hungary	12,240	4.9
Korea	27,195	5.8
France	37,675	6.5
Australia	50,962	7.3
United States	55,805	7.2

What is Corpus

A Corpus is a collection of Well-organized Data that has been developed or generated for solving a particular problem.

For example -

- IMDB 5000 Movie Dataset Corpus for Sentiment Analysis
- MNIST Corpus for English Digit Recognition
- ImageNet for Image Processing

Challenges of Machine Learning

The Main Challenges that can happen in Machine Learning are the following:

- Bad Data
- Bad Algorithm

BAD DATA EXAMPLES

☎ +88 0172 6867 984

🔗 tanvir@socian.ai

Insufficient Quantity of Training Data

For a small baby to learn what an apple is, you just point the Apple in front of the Baby and say the word “Apple” several times (possibly repeating this procedure a few times). Now the child is able to recognize apples in all sorts of colors and shapes.

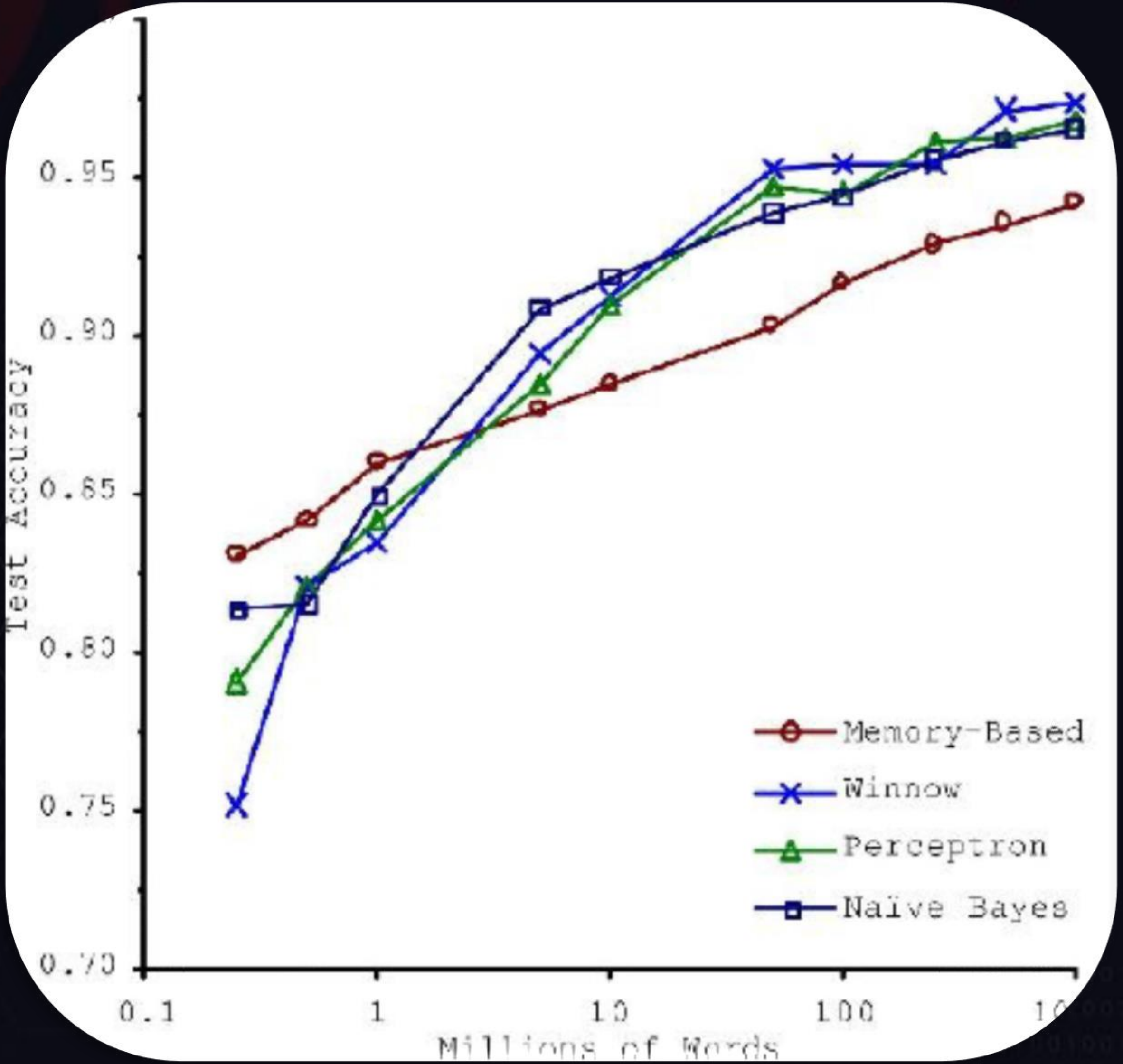
Genius!!!

Machine Learning is not quite there yet; it takes a lot of data for most Machine Learning algorithms to work properly. Even for very simple problems you typically need thousands of examples, and for complex problems such as image or speech recognition you may need millions of examples ***(unless you can reuse parts of an existing model)***.

Data is KING

In a famous paper published in 2001, Microsoft researchers Michele Banko and Eric Brill showed that very different Machine Learning algorithms, including fairly simple ones, performed almost identically well on a complex problem of natural language disambiguation once they were given enough data.

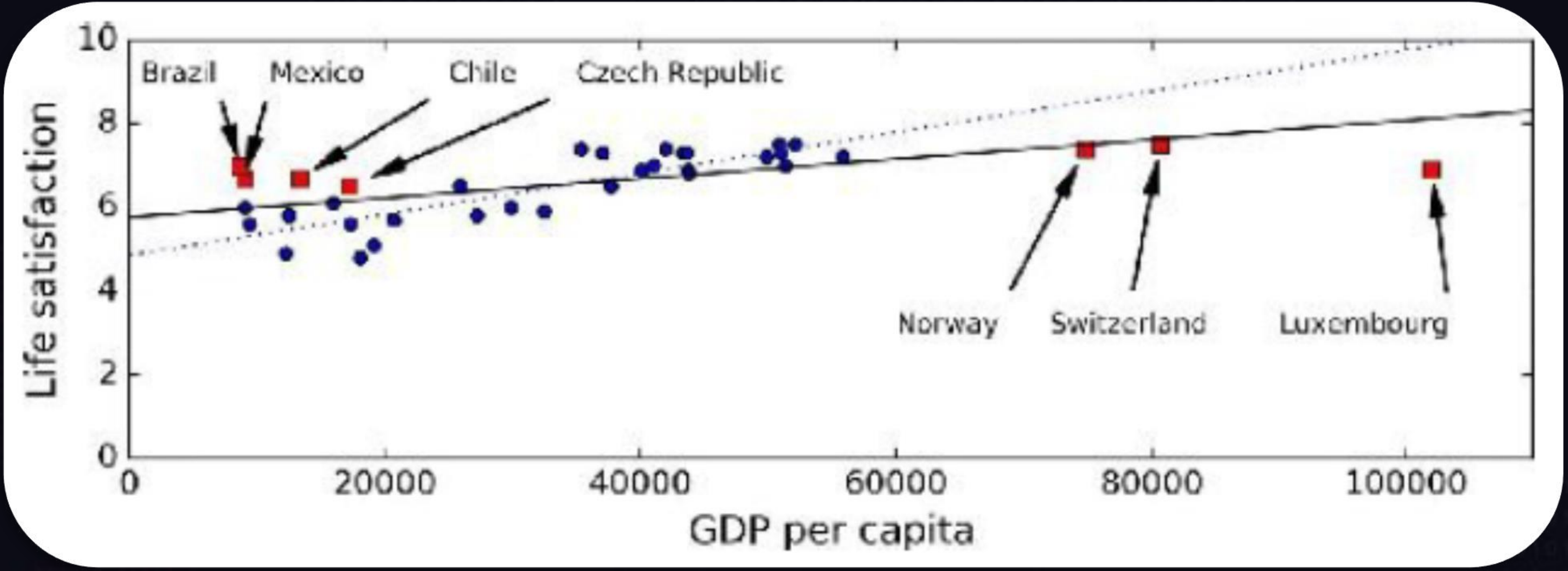
These results suggest that we may want to reconsider the trade-off between spending time and money on algorithm development versus spending it on corpus development.



Non-representative Training Data

It is very important that your Training Data can be properly represented by your Test Data that you want to generalize.

For example: If we look into the Missing Values for the use case where Money makes people Happy and we train a linear model on this data, we get the solid line, while the old model is represented by the dotted line:



Non-representative Training Data Cont.

On the previous figure, we can see that by adding a few missing countries significantly altered the model. From the figure, it makes it clear that such a simple linear model is probably never going to work well.

It seems that very rich countries are not happier than moderately rich countries (in fact they seem unhappier), and conversely some poor countries seem happier than many rich countries.

It is crucial to use a training set that is representative of the cases you want to generalize to. This is often harder than it sounds: if the sample is too small, you will have sampling noise (i.e., nonrepresentative data as a result of chance), but even very large samples can be nonrepresentative if the sampling method is flawed. This is called ***SAMPLING BIAS***.

Sampling Bias

Sample bias occurs when the distribution of one’s training data doesn’t reflect the actual environment that the machine learning model will be running in.

For example - If you’re trying to build a self-driving car, and you want it to drive at all times of day — at day and at night — and you’re only building training data based on daylight video, then that’s a bias that’s in your data.

The humans helping to build the training data for the algorithm can be completely correct, and have no bias. And yet the data is still biased because we didn’t include any nighttime examples. It’s bias to the day time.

It’s our job as data scientists to make sure the sample we’re building on matches the environment it’s going to be deployed in

Poor-Quality Data

If your training data is full of errors, outliers, and noise (e.g., due to poor-quality measurements), it will make it harder for the system to detect the underlying patterns, so your system is less likely to perform well. It is often well worth the effort to spend time cleaning up your training data.

The truth is, most data scientists spend a significant part of their time doing just that.

For example: You have obtained Voice Data collection with their transcription from a same user. Now, you want to built a Text to Speech System out of this data using AI. But unfortunately, 20% of these data contains noise.

The system will perform lot better if we can identify the audios that contain the noise and not use them in our Training.

Not Being Able to Identify Relevant Features

As we have seen over the time (The House Price Prediction example), we had lots of features in our available data. But all of these features may not be relevant for our System to learn. There is a saying: “Garbage in, Garbage out!”

Even the existing data may be missing some Relevant Features.

A critical part of the success of a Machine Learning project is coming up with a good set of features to train on. This process, called FEATURE ENGINEERING. It involves (We can get rid of this step in Deep Learning):

Feature Selection: Selecting the most useful features to train on among existing features.
Feature Extraction: Combining existing features to produce a more useful one (as we saw earlier, dimensionality reduction algorithms can help). Creating new features by gathering new data.