MACHINE LEARNING

by Dr Juan H Klopper

- Research Fellow
- School for Data Science and Computational Thinking
- Stellenbosch University

INTRODUCTION

Modeling is a very important aspect of Data Science. In the previous notebook, we were briefly introduced to linear models. Linear regression is a part of the larger topic of generalised linear models. Other than general linear models and similar techniques, we have machine learning.

In this notebook, we are introduced to the world of machine leaning. In the following two notebooks we will actually learn to implement to machine learning algorithms namely k nearest neighbours and random forests.

We begin our journey by introducing the machine learning as a sub type of a larger entity, general artifical intelligence.

PACKAGES USED IN THIS NOTEBOOK

```
1 import numpy as np
```

```
1 import plotly.graph_objects as go
2 import plotly.io as pio
3 pio.templates.default = 'plotly_white'
```

```
1 %config InlineBackend.figure_format = "retina" # For Retina type displays
```

A CLASSIFCATION OF MACHINE LEARNING

GENERAL ARTIFICIAL INTELLIGENCE

Simply stated, **general artificial intelligence** (AGI) is the ultimate goal of our pursuit in artificial intelligence. It refers to an electronic or non-human system with the ability to learn and understand at the level of a human being. Such an entity would be able to communicate, create, experience sentience and self awareness.

Our machines are far from this goal. Instead, we have been working on artificial intelligence (AI). This subset of AGI (or stepping stone towards AGI) merely refers to the combination of software and hardware to solve specified tasks.

There are various types of AI. We will briefly discuss expert systems, probablistic systems, and functional systems.

EXPERT SYSTEMS

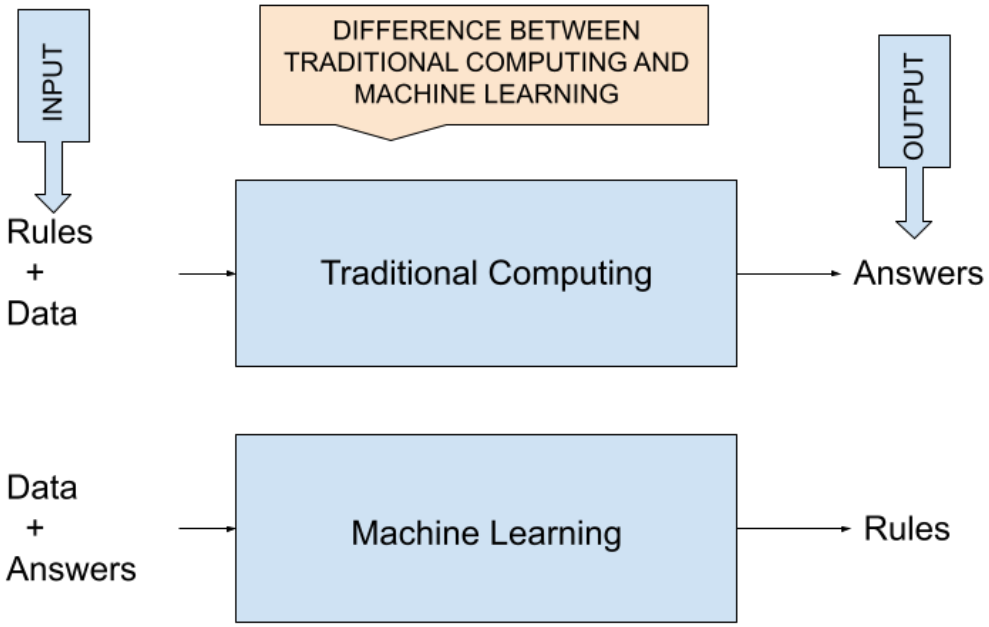
This older approach to AI refers to rule based systems. The idea that if we were to program a computer with enough rules, it will know how to solve a problem.

Here, we have to create specific code that a program must follow to solve the probelm. This is much harder than you can imagine. Think of identifying objects in an image. The number of rules that we would ahve to create are enormous. This is also not how our human minds learn new tasks. We are much more adept to learn from data.

An example of these systems include the previous generations of voice recognition software. These have been replaced by neural networks that are much more robust.

FUNCTIONAL SYSTEMS

This is a mathematical approach to AI. We use functions that mimic some aspects of learning. Instead of providing the rules, we provide the data and the solution and let the machine calculate the rules. This is demonstrated in the image below.



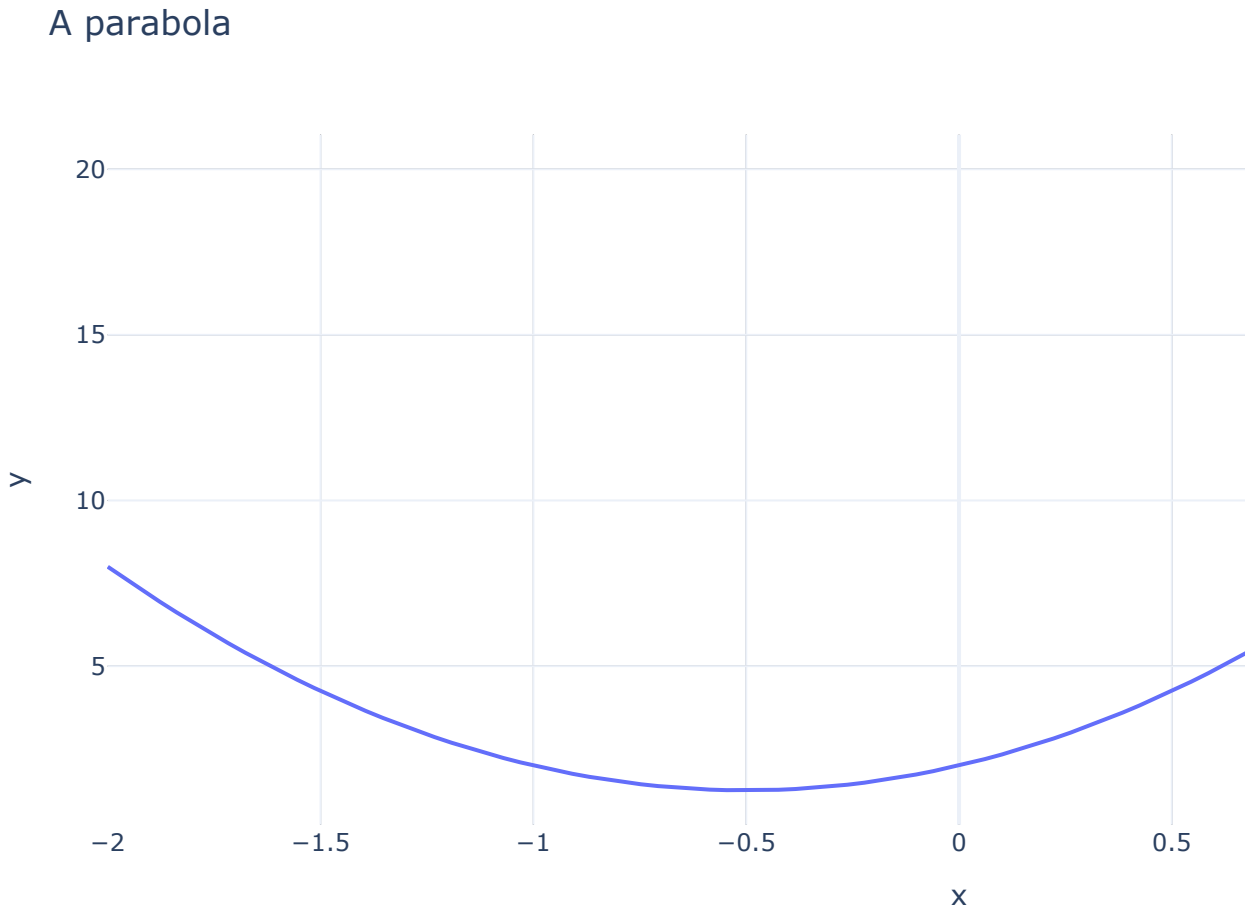
In neural networks (a type of AI) we generate loss functions. These measure the difference between a known ouctome and the ouctome predicted by the neural network. These prediction dependend on changable values called parameters. During the *learning* process the neural network adapts these paramaters to diminish the loss function with the result that predicted values and the actual values are much closer to each other. In other words, it learns to be more accurate in its predictions.

We are actually famliar with this approach. Remeber the parabola from school algebra. We can use it as a surrogate for a loss function.

Below, we generate some values to plot the function $3x^2 + 3x + 2$.

```
xvals = np.linspace(-2, 2, 100)
yvals = 3 * xvals**2 + 3 * xvals + 2
```

```
go.Figure(
    go.Scatter(
        x=xvals,
        y=yvals,
        mode='lines'
    )
).update_layout(
    title='A parabola',
    xaxis={'title':'x'},
    yaxis={'title':'y'}
)
```



With a parabola, its is quite easy to *see* where the minimum of this function is. With minimum we refer to the x value for which the y value is at a minimum.

In reality a loss function is much more complex, with many more dimensions. The point, though, is that in a neural network, values for x and many other parameters are calculated that will give us a minimum. The network *learns* these best values.

Mathematics is an excellent tool to get machines to learn. In the next two notebooks on k nearest neigbours and random forests we will see other mathemtical functions working behind the scenes to *learn* the best results.

MACHINE LEARNING

Machine learning (ML) is then a subtype of AI. There are numerous subtypes of ML. We have already mentioned deep neural networks, k nearest neighbours, and random forests.

Irrespective of the ML subtype we employ, we need to classify the problem we want to solve before selecting the best ML subtype.

Here, we look at the types of ML problems and mention three.

SUPERVISED MACHINE LEARNING

The essence of this type of ML is the presence of a known *target variable*. Consider tabular data where we have independent variables (referred to as feature variables in ML) and a dependent variable (referred to as a target).

It is common in ML to split our data into a training and a test set. An ML algorith learns from the training set and because we know the target variable value in the data, we can verify metrics such as the accuracy of the ML model against the known values in the target variable.

There are two types of supervised ML problems. The type depnds on the data type fo the target variable. If it is categorical were refer to the problem as a **classification problem**. If the target variable type is continuous numerical, we refer to a **regression problem**.

UNSUPERVISED LEARNING

In unsupervised learning we do not have a target variable. Here, the obnservations are clustered by an algorith such that certain observations cluster together. We have to put meaning to the clusters.

REINFORCEMENT LEARNING

In this type of ML the machine learns from a reward-penalty system. It learns to navigate the world by amassing rewards and minimising penalty. This is how computer learn to beat chess and go champions.

CONCLUSION

This was a brief introduction to ML. In the next two notebooks we learn more about k nearest neighbours and random forests.

1

✓ 0s completed at 14:06