

Practical Machine Learning



Practical Machine Learning

Lecture: Introduction to Machine Learning

Ted Scully

Lecturer - Ted Scully

- Lecturer in Computer Science
 - Joined CIT in 2013
 - **Programme Coordinator** for MSc in AI
 - Principal Investigator in Riomh Research Group
 - Associate Investigator with SFI CONNECT
 - PhD Students (3 Students)
- AI Research and Development Experience
 - Machine Learning
 - Deep Learning
 - Meta-heuristic optimization



Selected Research Projects

- Real-time Detection of Weather Events with Distributed Machine Learning for SmallSat Platforms.
- A framework for PU Learning with GANs.
- Demand side management of water and energy consumption in the Irish dairy industry.
- Optimal battery utilization for micro-grid cost minimization.



Class Rep

- If anyone is interested in acting as a class rep please send me an email with the title Class Rep (ted.scully@cit.ie)

Electives

- You need to decide on your chosen elective before **Friday Oct 9th**.
- NLP
 - Students can self-enrol in the module once you share with them this URL: <https://cit.instructure.com/enroll/FT4L8A> . Alternatively, they can sign up at <https://cit.instructure.com/register> and use the following join code: FT4L8A
- Distributed Ledger Technology
 - This module has enabled open enrolment. Students can self-enrol in the module once you share with them this URL: <https://cit.instructure.com/enroll/DFE76L> . Alternatively, they can sign up at <https://cit.instructure.com/register> and use the following join code: DFE76L
- Programming Language Design
 - This module has enabled open enrolment. Students can self-enrol in the module once you share with them this URL: <https://cit.instructure.com/enroll/A7DRRW>
 - Alternatively, they can sign up at <https://cit.instructure.com/register> and use the following join code: A7DRRW
- Software Engineering Processes
 - This module has enabled open enrolment. Students can self-enrol in the module once you share with them this URL: <https://cit.instructure.com/enroll/JW8LBJ> . Alternatively, they can sign up at <https://cit.instructure.com/register> and use the following join code: JW8LBJ

Practical ML - Course Breakdown and Assessment

- Email: ted.scully@cit.ie
- Weekly Schedule
 - Lectures 2*1hr
 - Discussion Forum
 - Practical lab (commencing in Week 2)
- Module is 100% Continuous Assessment.
 - Project 1 - Develop a machine learning model for a real-world problem and perform a comprehensive analysis. (50%).
 - Project 2 - Perform a comparative analysis of a range of machine learning classification algorithms applied to a dataset from an application domain. (50%).
 - Timing matrix for assessments will be sent out over the next few days

Content

- Pre-processing
 - Application of pre-processing techniques such as outlier detection, feature selection, imputation of missing data, encoding, normalization, deal with imbalanced datasets etc.
- Evaluation and Model Selection
 - Best practice evaluation techniques such as precision, recall, confusion matrices and ROC curves. Debugging algorithms using validation and learning curves. Cross fold validation. Model selection using hyper parameter optimization.
- Classification Algorithms
 - Classification algorithms such as decision trees, ensemble technique (bagging and boosting, gradient-boosting), instance-based algorithms, naïve bayes, Bayesian networks, etc.
- Unsupervised Algorithms
 - Overview of unsupervised learning techniques. Example applications of clustering techniques. Introduction to algorithms such as k-means, k-median, dbscan and hierarchical clustering techniques. Optimization and distortion cost function. Random initialization and methods of selecting number of clusters. Silhouette plots.

Resources

We assume that everyone starting this course is able to code in Python and has a reasonable grasp of NumPy and Pandas.

We will be using Python 3 (preferably 3.8) as our programming language in this module.

- The following are basic tutorials to get you up and running with the syntax and control structure for Python.
- [Python 3 Tutorial](#) – Clear and focused overview of Python 3 syntax, control structures, data structures etc.
- [Video Python 3 Tutorials](#) – A set of very basic Python 3 video tutorials. More focused on beginners.
- [Automate the Boring Stuff with Python](#) - Learn to Code. If you've ever spent hours renaming files or updating hundreds of spreadsheet cells, you know how tedious tasks like these can be.
-
- The following are a number of accessible tutorials to help you get started with NumPy and Pandas.
- [DataCamp NumPy Tutorial](#) – Accessible and easy to understand tutorial to get started with NumPy
- [NumPy Tutorial](#) – Short overview of NumPy and basic Python data structures. It also covers SciPy (which you don't need) and basic Matplotlib (which you will be covering later in the programme as part of visualization).
- [DataCamp Pandas Tutorial](#) – Short and easy to understand tutorial on using Pandas

Resources - DataCamp Access

I have applied for academic access to DataCamp and you will be receiving an email invitation, which will grant you access.

Please let me know if you don't receive this email before the end of week 1.



Resources

- **Websites**

- [Machine Learning Stanford](#) – Andrew Ng
- [Machine Learning Class \(Washington\)](#) - Pedro Domingos
- [Udacity Machine Learning](#) - Sebastian Thrun
- [UCI Data Repository](#)
- [Kaggle](#)

- **Books**

- [Hands-On Machine Learning with Scikit-Learn and TensorFlow](#) – Aurelien Geron (2nd Edition)
- [Python Machine Learning](#) - Sebastian Raschka (3rd Edition)
- [Fundamentals of Machine Learning for Predictive Data Analytics](#) – (John Kelleher, Brian MacNamee , Aoife D'Arcy)

Software Options (1. Remote Desktop)

- Under normal circumstances the MSc in AI get swipe card access to a practical lab C127 (all machines in the lab have a GTX 1080 GPU).
- Computer services are currently configuring this lab for remote access.
- The IT Services are working very hard to have the machines in these labs available for remote login by the end of Week 1 of term.
- Once available I will circulate information on how to access.
- There are **different environments** on each machine which you can enable for different module requirements ... more on this later!

Software Options (2. Anaconda)

- Anaconda is an open-source distribution of Python.
- It comes with a range of essential packages such as NumPy, Pandas, Scikit-Learn and Matplotlib, TensorFlow, PyTorch.
- Spyder IDE or Jupyter Notebook.
- Conda - an open source package management system (isolated environments).
- Download [Anaconda](#) (Python 3.8 version)

Anaconda Installers

Windows

Python 3.8

64-Bit Graphical Installer (466 MB)

32-Bit Graphical Installer (397 MB)

MacOS

Python 3.8

64-Bit Graphical Installer (462 MB)

64-Bit Command Line Installer (454 MB)

Linux

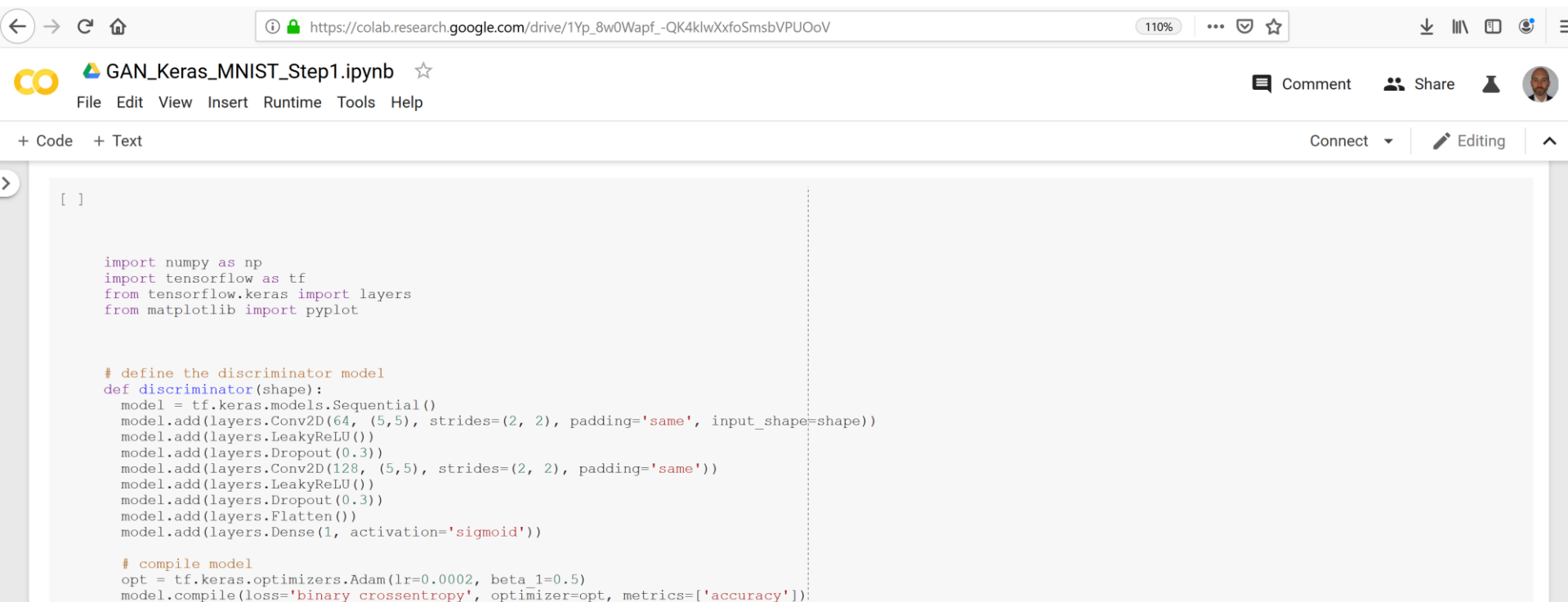
Python 3.8

64-Bit (x86) Installer (550 MB)

64-Bit (Power8 and Power9) Installer (290 MB)

Software Options (3. Colab)

- [Google Colab](https://colab.research.google.com/) is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud.
- Again comes with essential packages such as Scikit-Learn, NumPy, etc all pre-installed.
- Comes with the option of a free GPU or TPU (not necessary for Practical ML Module)
- **Drawback**. When you connect to a VM runtime, you have a maximum of 12 hours on the VM. You can easily connect to another VM after the 12 hours expires but you will lose access to an data you had in the previous VM.



The screenshot shows a Google Colab notebook titled "GAN_Keras_MNIST_Step1.ipynb". The interface includes a browser address bar with the URL "https://colab.research.google.com/drive/1Yp_8w0Wapf_-QK4klwXxfoSmsbVPUOoV", a toolbar with icons for navigation and settings, and a menu bar with options like File, Edit, View, Insert, Runtime, Tools, and Help. The notebook content shows a Jupyter cell with the following Python code:

```
[ ]

import numpy as np
import tensorflow as tf
from tensorflow.keras import layers
from matplotlib import pyplot

# define the discriminator model
def discriminator(shape):
    model = tf.keras.models.Sequential()
    model.add(layers.Conv2D(64, (5,5), strides=(2, 2), padding='same', input_shape=shape))
    model.add(layers.LeakyReLU())
    model.add(layers.Dropout(0.3))
    model.add(layers.Conv2D(128, (5,5), strides=(2, 2), padding='same'))
    model.add(layers.LeakyReLU())
    model.add(layers.Dropout(0.3))
    model.add(layers.Flatten())
    model.add(layers.Dense(1, activation='sigmoid'))

# compile model
opt = tf.keras.optimizers.Adam(lr=0.0002, beta_1=0.5)
model.compile(loss='binary_crossentropy', optimizer=opt, metrics=['accuracy'])
```

Software Options (3. Colab)

- To use Colab you will need a Google (GMail) account.
- Another aspect of Colab is that you can **mount files from your Google Drive**. This allows you to easily access data from the Colab VM instance.
- I have included a short guide to getting started with Google Colab in the Week 1 unit on Canvas.
 - Describes how to create a Colab Notebook from your Google Drive.
 - Mount a data file
 - Open the data file and perform some basic pre-processing on the data file.

Machine Learning

1. Machine Learning is a important **sub-discipline of AI**.
 2. One goal of AI is building programs to perform tasks, which **humans are currently better at**. Machine learning is an avenue that has had success doing exactly that.
- How do you program a computer to:
 - Recognize faces?
 - Identify objects in images
 - Interpret hand written text
 - Interpreting spoken language?
 -

Machine Learning

How do we define Machine Learning?

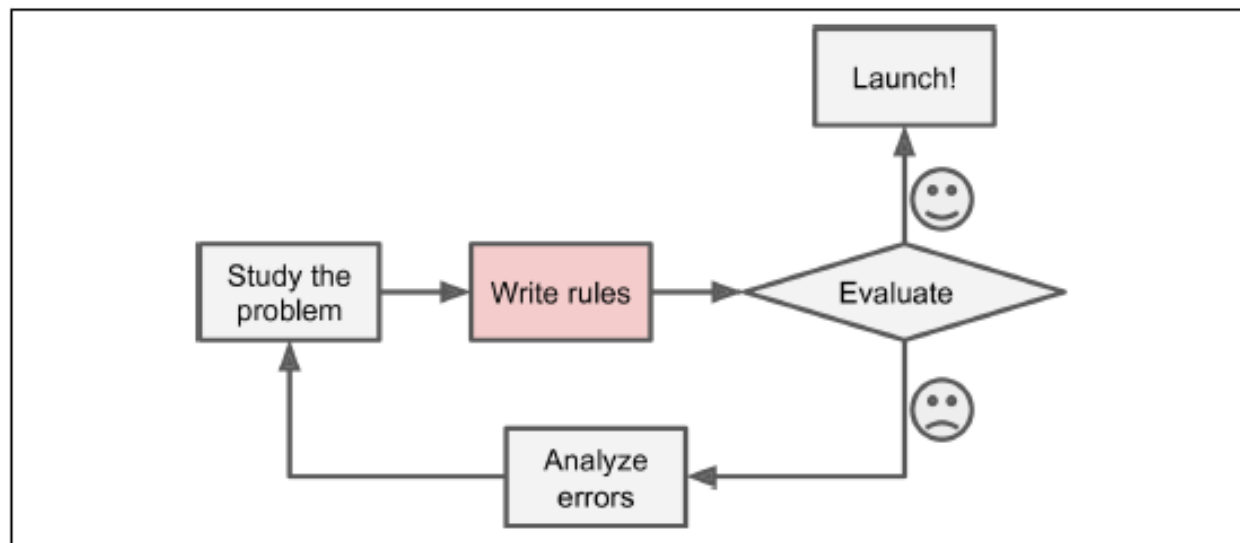
A computer program that will learn from experience E with respect to some task T and some performance measure P , if its performance on T , as measured by P , improves with experience E .

—Tom Mitchell, 1997

Machine learning (ML) provides a means by which programs can infer new knowledge from observational data.

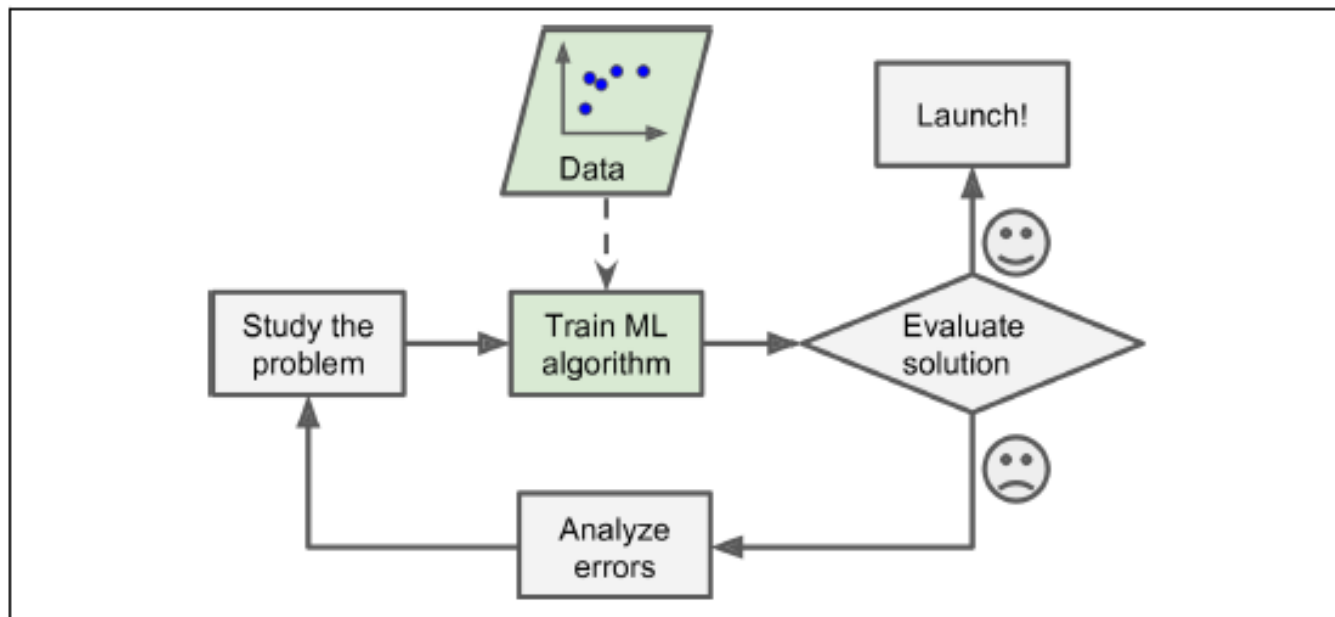
Why Use Machine Learning?

- Let's take a basic problem such as building a **spam filter**.
- We could attempt to build a spam filter using traditional programming techniques.
- First you would look at what spam typically looks like and observe that certain words tend to occur quite frequently in spam.
- You would write a detection algorithm for each of the patterns that you noticed, and your program would flag emails as spam if a number of these patterns are detected.



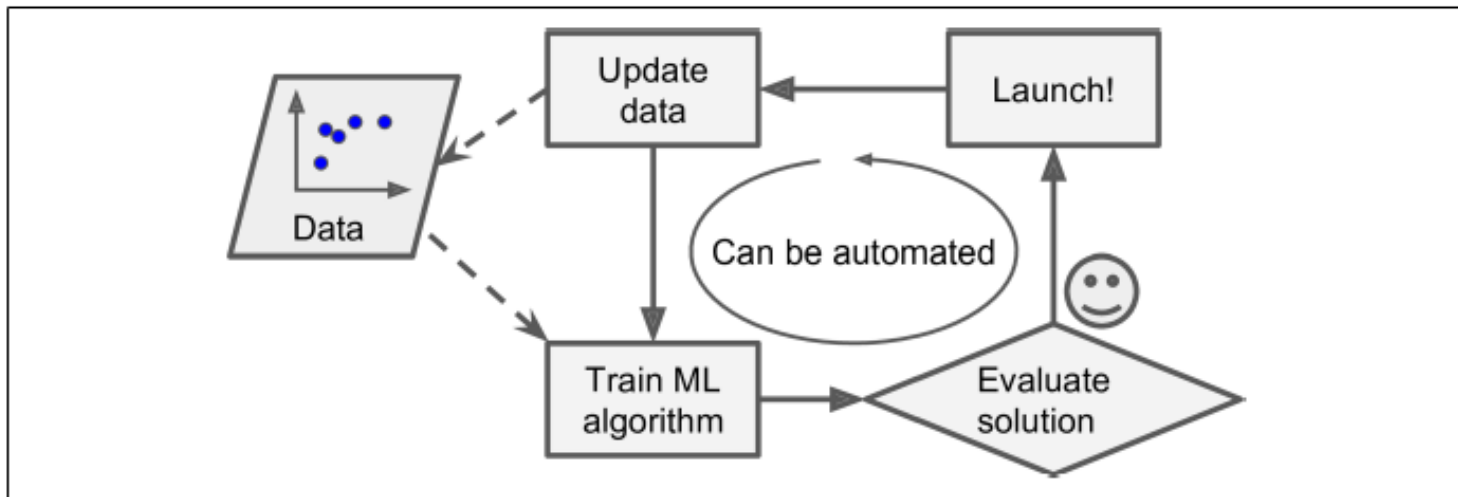
Why Use Machine Learning?

- In contrast, a spam filter based on Machine Learning techniques automatically learns a model (by looking at the words and phrases that are good predictors of spam).
- The program is much shorter, easier to maintain, and most likely more accurate.



Why Use Machine Learning?

- Building such a system using machine learning also means that we can easily **update our model**.
- It is often necessary to retrain models periodically. This is particularly important in scenarios where there is drift in the data over time.



Machine Learning Applications



Spam

- Host of machine learning algorithms that will learn to classify emails as spam.



Natural Language Processing

- Speech recognition
- Machine Translation

Machine Learning Applications



Manufacturing and Robotics

- Manufacturing – Quality inspection, predictive maintenance, etc
- Robotics - Recognition of objects , navigation of spaces, etc.



Commercial/Finance

- Applications include trading agents that interact with the bond, stock or commodity markets.
- Sentiment Analysis
- Forecasting and Prediction

Machine Learning Applications



Navigation

- Research in self-driving cars goes back to early 1990's.
- From Alvin and Stanley (212 km course, 2005) to Google's Self Driving Car.



Recommender Systems

- Netflix, Amazon, Google all use recommender systems
- Collaborative and Content Filtering
- Marketing

Machine Learning Applications



Games

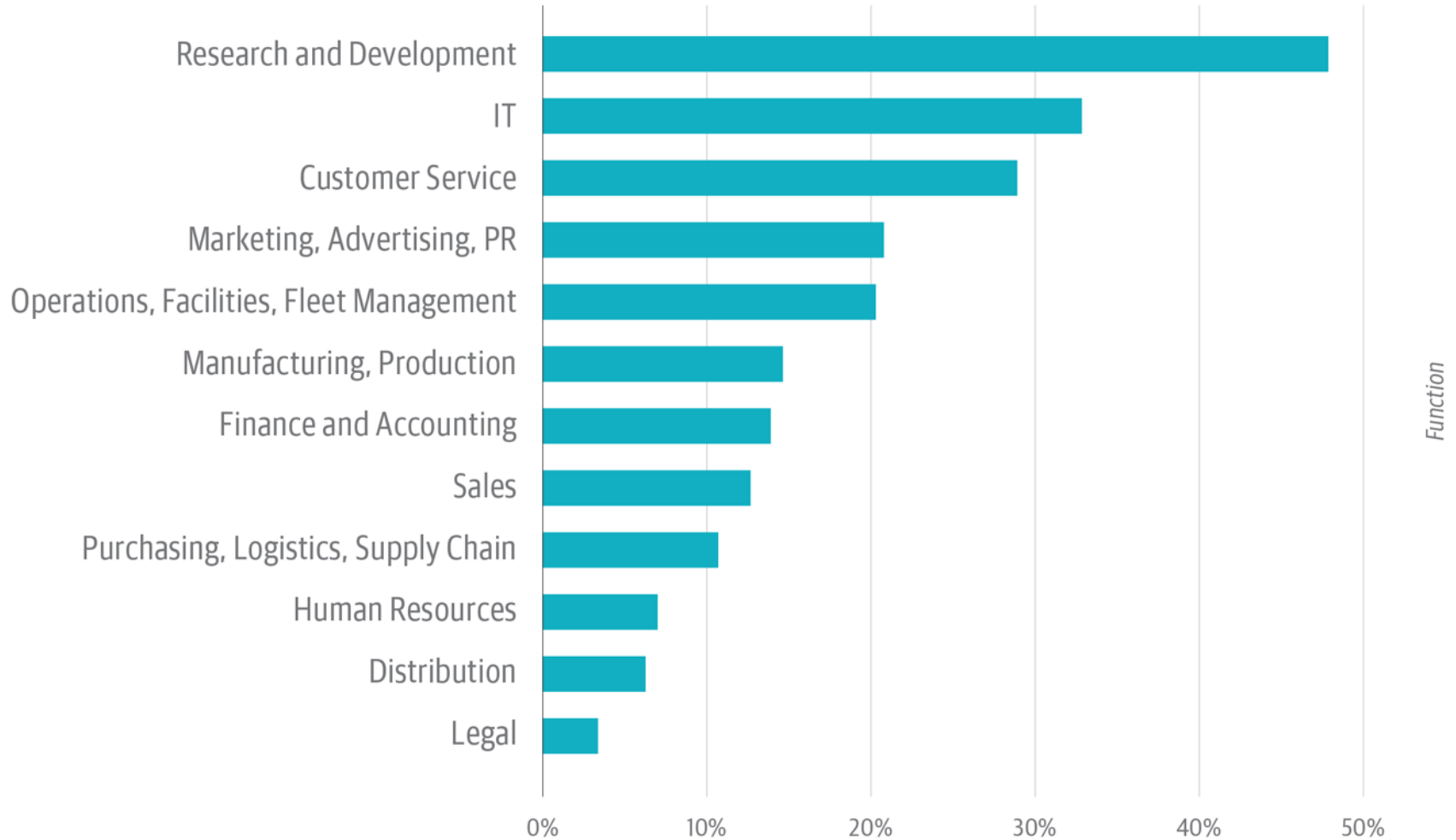
- Machine Learning has proved successful in its application to gaming from Arthur Samuel's checkers program to IBM's Watson and Alpha Go.



Medicine

- Medical applications can provide decision support systems for assisting in the diagnosis of patients or identification of particular illnesses.

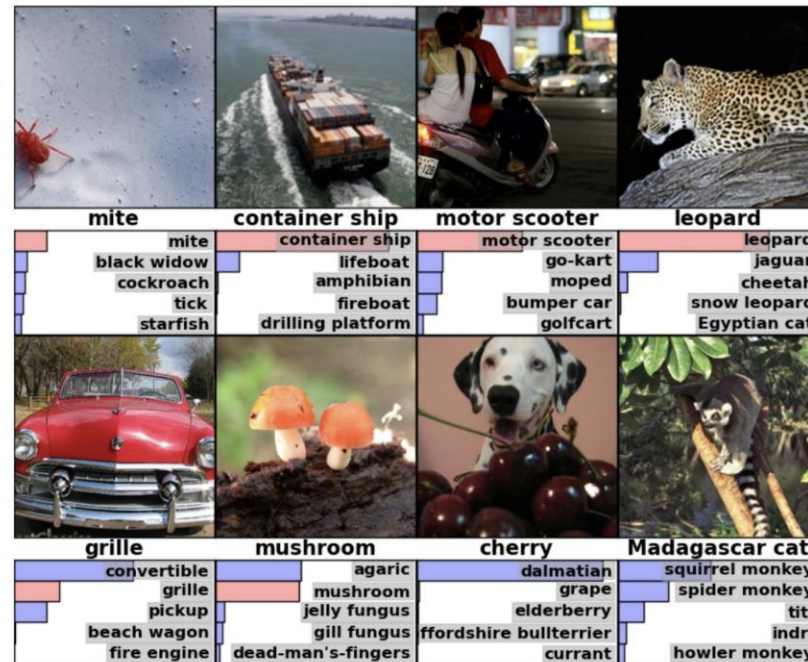
In which functional parts of the company are AI projects used?



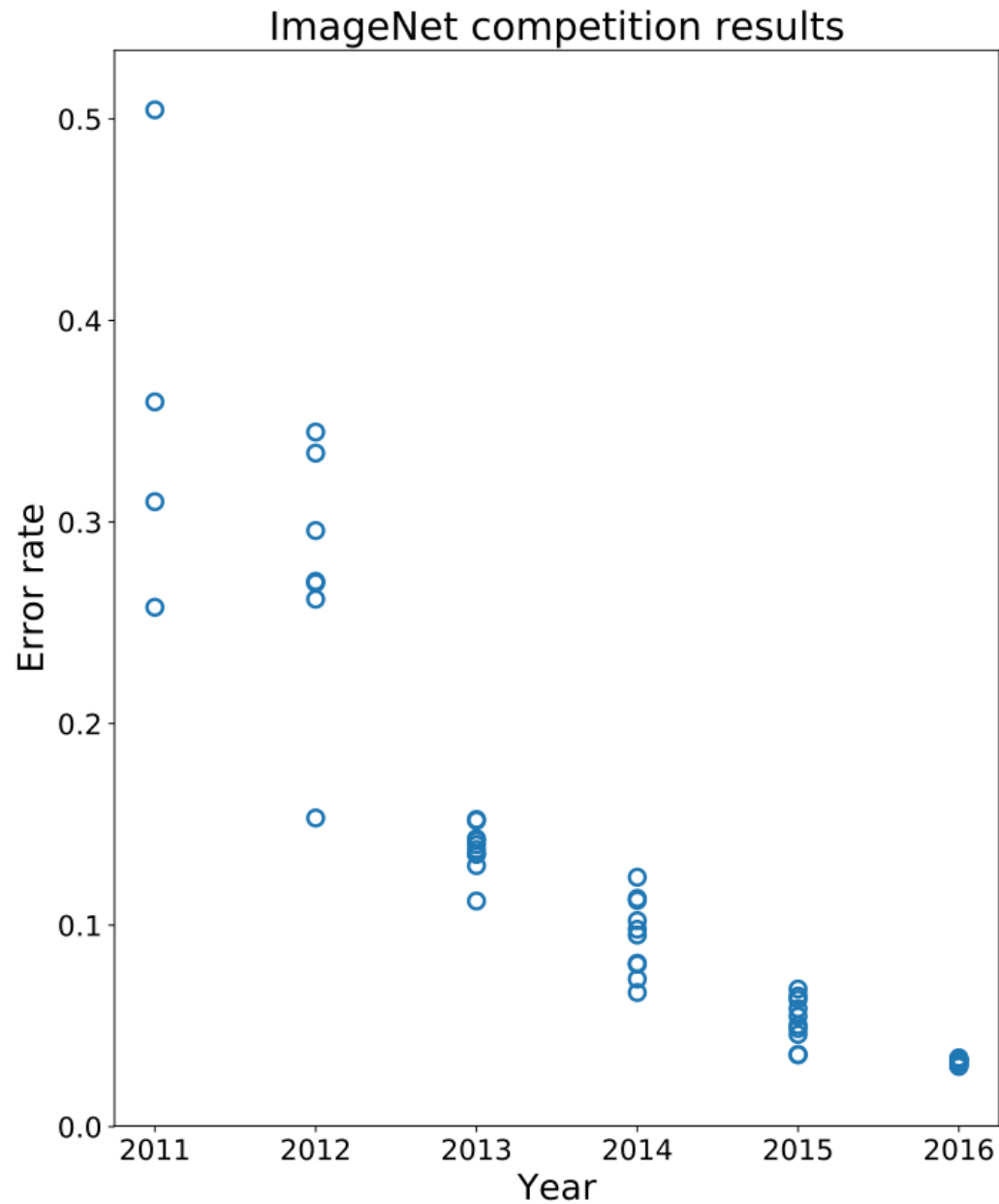
ImageNet Challenge



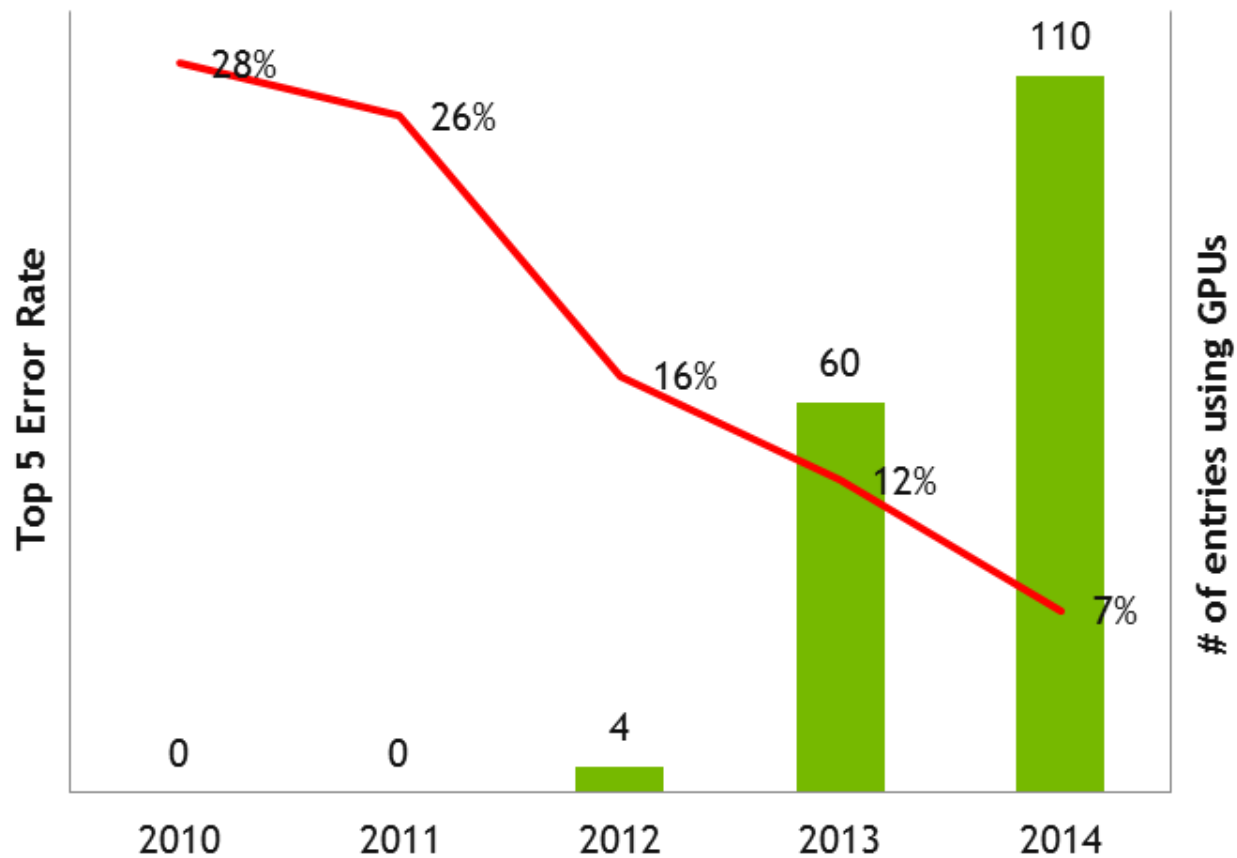
- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



- The annual ImageNet competition began in 2010 where software programs compete to correctly classify and detect objects and scenes.
- In 2012 a submission called **AlexNet** achieved a **top-5 error of 16%**, more than 10.8 percentage points ahead of the runner up.
- As of 2020 it has been cited over 69,990 times.



IMAGENET



ImageNet Challenge

- GoogLeNet (also called Inception V1) won the ImageNet competition in 2014.
- ResNet won the ILSVRC 2015 competition with an incredible 3.6% error rate (human performance is 5-10%).
- In 2017, 29 of 38 competing teams got less than 5% wrong.

