**ML Applications**
- Optical Character Recognition
- Spam Detection
- Fraud Detection
- Customer Service/Chat bots
- Recommendation/Personalisation
- Information Extraction
- Machine Translation
- Autonomous Driving
- Speech Recognition
- Face recognition
- Medical Diagnosis
- Employee Access Rights
- Management (ML Employee task assignment)
- Compose Music
- Create and Extend Art

**Outline**
1. Traditional Problem Solving Approaches
2. ML Approach to Problem Solving
3. Different Types of ML
4. An Example of how ML works
5. Strengths and Weaknesses of ML
6. Context of ML (AI; Data Science; Big Data; ...)

**1. Traditional Approaches to problem solving**
- Human Labour
- Brute Force
- Rules/Heuristics
    - EXAMPLE: Spam Detection
        - Goal: Decide if an email is spam or not? (**classification problem**)
        - Potential Rule: identify spam terms (porn, sex, p?rn, sex*, etc..)
        - But what happens when new wildcards get through? P0rn
        - **Cyclical process** - revise, test, fail… tedious and time intensive
    - EXAMPLE: Document Similarity
        - Goal: Determine how similar 2 documents are
        - Absolute overlap: total number of shared features (word count etc)
        - Jaccard Index: relative number of shared features
            - $J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A|+|B|-|A \cap B|}$
            - Jaccard Distance = 1-Jaccard Index
        - **Vector Space and TF-IDF Weighting**
            - Document represented as a term vector in an n-dimensional space (n = number of different terms)
            - Term weights can be done by frequency, and similarity is the cosine distance between 2 vectors

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|_2 \|\mathbf{B}\|_2} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} \sqrt{\sum_{i=1}^{n} B_i^2}}$$

- More advanced term weighting: TF-IDF

$$TF - IDF = tf(t) * \log \frac{N_r}{n_r}$$

**2.ML approach to problem solving**

- *"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."*
- EXAMPLE: Approach to Spam Detection
    - Given the above quote.. T=Detecting Spam Emails (flag new emails as spam/not spam), P=number of valid classifications, E=Dataset with emails classified as spam/not-spam
    - Study Problem -> Train on Data -> Eval Solution -> Analyze Errors -> Repeat Until Launch Ready
    - **You update the data, not the algorithm** and learning can be automated
- New insights from ML e.g. Data mining
    - Discover which words indicate if an email is spam or not - difficult for some algorithms
- **Formalized**
    - Goal 1: Prediction
        - Find a function f that can predict Y given X, with X = x_1, x_2,...,x_n
        - Predict Y for x_new
    - Goal 2: Inference
        - Understand which variables affect Y
    - $Y = f(X) + \varepsilon$ ⇔ Target = Model applied to explanatory features x1->xn + noise
- **ML as a general problem solver**
    - **One algorithm can solve many different problems**
        - E.g. SVMs can solve Spam Detection, OCR, Image classification
    - **No one algorithm can do everything**

**3.Types of ML**

- **Supervised Learning**
    - **Typical Tasks**
        - **Classification**
        - **Prediction/Regression**
    - **Training Input**
        - **Labelled Data**
        - Instances/Data Points with known values for independent variables <u>and</u> known dependent variables/label
    - **Typical Algorithms**
        - **K-Nearest neighbour**
        - **Regression**

        - **Support Vector Machines**
        - **Decision Trees and Random Forests**
    - Production
        - Input: new instance with known independent variables
        - Output: Predicted target/label
- **Unsupervised Learning**
    - Unlabelled data
        - **Work on the data you have**, don't predict the future
    - Typical Tasks
        - **Clustering into groups** which may not have been clear
        - **Dimensionality reduction**
    - **Algorithms**
        - **Clustering**
            - **K-means**
            - **Hierarchical Cluster Analysis (HCA)**
            - **Expected Maximization**
        - **Dimensionality Reduction**
            - **Principal Component Analysis**
            - **Kernal CPA**
- **Semi-Supervised Learning** (less common)
    - **Some labelled but mostly unlabelled data**
    - **E.g. Google Photos**
        - Identify faces and cluster (unsupervised) and labels are added by user to clusters (supervised)
    - **Often based on Deep Belief Networks (DBN)/Restricted Boltzmann Machines (RBM)**
- **Reinforcement Learning** (also less common)
    - An **agent observes an environment**, **chooses an action** and is **rewarded or penalised**, then **best strategy is determined by historical data** given the situation
    - Used in making walking robots and DeepMind's AlphaGo
- **Batch Learning vs Incremental Learning**
    - **Batch/Offline Learning**
        - **Trained on all data**, **when new data** is available must be f**ed old and new**
        - Can be automated but **costs a lot** of time and resources
    - **Incremental/Online learning**
        - **New/Updated training is done on new data only**
        - Updates occur sequentially, **Data can be disposed of after training**
        - **Good for** systems with:
            - **Continuously new data**
            - **Limited computing resources**
            - **Huge volumes of data**
- **Model vs Instance-based Learning**
    - Describes how you 'generalize'
    - **Model-based**

- **Find a function that describes** the existing instances as accurately as possible
        - **Calculate** the value for the new instance **based on the function**
    - **Instance-based**
        - **Find instances similar** to the new one
        - **Assume that the values** of similar instance **also apply** to the new one
- **Parametric vs Nonparametric Models**
    - **Parametric**
        - **Based on the features, and certain weights, Y can be calculated**
        - **E.g. (Linear) Regression**
            - $f(X) = \beta_0 + X_1 * \beta_1 + X_2 * \beta_2 + \ldots$
    - **Nonparametric**
        - **E.g. A decision Tree**
        - **No fixed functional form**, can **grow in complexity** with large amounts of data **to capture complicated patterns**
- **5 tribes of ML**
    -

| **Tribe** | **Origins** | **Problem** | **Solution/Master Algorithm** |
|---|---|---|---|
| Symbolists | Logic, Philosophy | Knowledge Composition | Inverse Deduction |
| Connectionists | Neuroscience | Credit Assignment | Backpropagation |
| Evolutionaries | Evolutionary Biology | Structure Discovery | Genetic Programming |
| Bayesians | Statistics | Uncertainty | Probablistic Inference |
| Analogizers | Psychology | Similarity | Kernel Machines |

- **Hill Climbing Example**
    - **Iterative Algorithm**
    - **Start** with **arbitrary solution - incremental improvements**
        - **Repeat until no improvement** is achieved
    - **Disadvantage:** Find only the **local optima & computing intensive**

**4. Strengths and Weaknesses of ML**

- ML is **good for** problems…
    - Which need **large rule bases**
    - Which are **too complex** to be solved with heuristics
    - That have **changing environments/data**
    - Where **data is already available**
- **But…**

- **Lots of data needed**
- Data is **not always available**
    - No data no ml
- **Blackbox**
    - **Not always clear** why an output it generated
    - **Understanding** the reasoning behind the **output requires** a lot of **thought**
    - Legal implications - Google Antitrust (Google fined for pushing their service over others)
- **Can't unit test**

**5. Friends of ML**

- AI
    - Broader than ml
        - Natural language processing, simulation, robotics, pattern recognition
    - Not all AI is ML, not all ML is AI
- Deep Learning
    - Subset of ML
    - "DL is 1% of ML  which is 1% of AI"
    - Can detect more abstract features - useful in image/speech recognition
- Data science
    - Focus on data, not learning
    - Data integration, distributed architecture, automated machine learning, Data visualization, Dashboards and BI, data engineering etc etc
- Big Data
    - Volume, Velocity, Variety, Variability, Complexity
        - https://www.sas.com/en_us/insights/big-data/what-is-big-data.html
- Computational Intelligence
    - Solving optimization problems + machine learning
    - Three branches
        - Evolutionary algorithms
        - Neural Networks
        - Fuzzy Logic