Tech Trek: Navigating the Landscape of Modern Technologies and Data Science

Premanand S

Assistant Professor
School of Electronics Engineering (SENSE)
VIT University - Chennai Campus

October 14, 2023

Why is this Tech-talk?!

Technology

- Technology refers to the application of scientific knowledge, tools, techniques, and systems to solve problems, improve processes, achieve specific goals, or create new products or services.
- Technology can be physical, like machines or devices, or conceptual, like algorithms and software.

Why Technologies?

- Solving Problems
- Improving Efficiency
- Enhancing Communication
- Security and Defence
- Quality of life
- Advancing Science
- Still many reasons...

Trending Technologies

- AI, ML, DL...
- 5G Technology
- IoT
- Blockchain
- Quantum Computing
- Cybersecurity
- AR and VR
- Robotics and many more...

THE Technologies

- Artificial Intelligence
- Machine Learning
- Deep Learning
- Data Science
- Natural Language Processing
- Computer Vision

Are we surrounded by AIR?

We are surrounded by DATA, but starved for INSIGHTS!

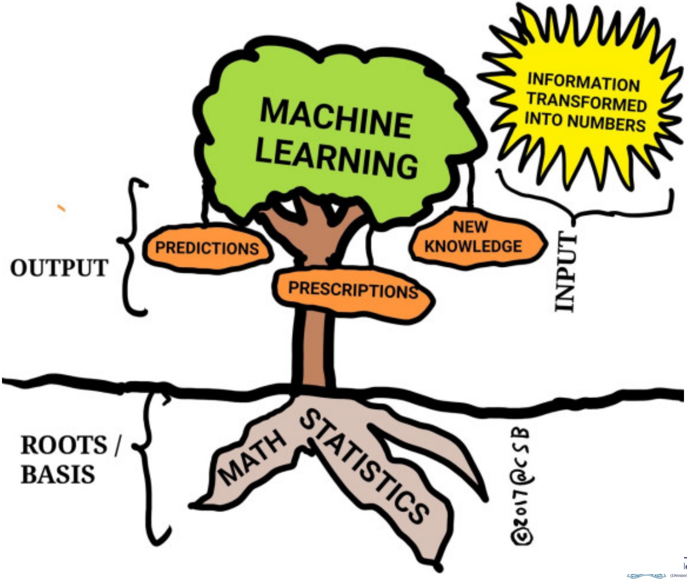
Watch the Video

Click to Watch

Statistics

- Statistics is the science of collecting, organizing, analyzing, interpreting, and presenting data
- Data Collection
- Descriptive Statistics
- Inferential Statistics
- Data Visualization
- Probability
- Statistics in data science is your secret tool for understanding data, making predictions, and checking if your ideas are correct.
- It's the magic that turns a bunch of puzzle pieces into a clear picture, helping us make better decisions and discoveries.

Is Statistics essential for Machine Learning?



Importance of Statistics

- Statistics - Problem - Dataset
- Machine Learning - Dataset - Solution

Machine Learning - Intro

- **General Intro** Machine Learning, means it can access the data and use it to learn for itself without any programming.
  "Machine Learning is the field of study that gives computers, the ability to learn without being explicitly programmed. — **Arthur Samuel, 1959**"
- **Engineering Intro** - 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. — **Tom Mitchell, 1997**
- Machine Learning - Animation @Simplilearn

Real-time examples for Machine Learning

- Facial Recognition
- Virtual Reality headsets - MIT
- Speech to text (iPhone users)
- Robo dog - Spot! - Spot - Dance for Uptown! (Reinforcement Learning)
- Amazon, Flipkart, Netflix, Audible (E-Commerce) (Recommender System)
- Weather, Stock-market analysis, Medical Images and Signals (ECG, PPG, EEG, EMG...) (Time Series)

Types of Machine Learning

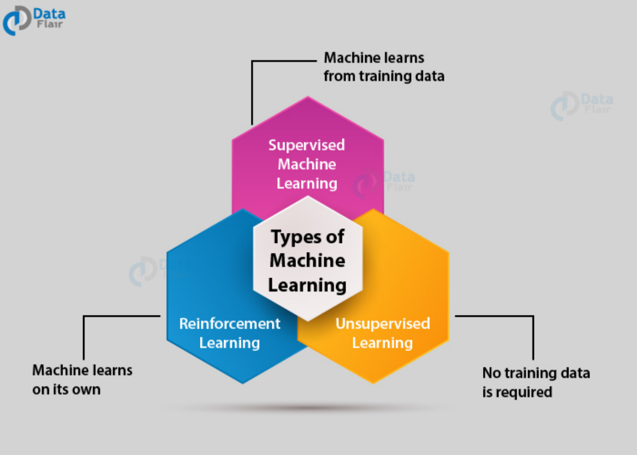


Figure: Broad classification of ML

Types of Machine Learning (Cont...)

- **Supervised Machine Learning** - Train me!
  - Classification
  - Regression
- **Unsupervised Machine Learning** - I am self sufficient in learning!
  - Clustering
  - Dimensionality Reduction
  - Association Rule
- **Reinforcement Learning** - My life My rules!
- **Semi-Supervised Learning**

Algorithms for each type of Machine Learning

Supervised Machine Learning - Classification

- Support Vector Machine (Linear SVM)
- Kernel Support Vector Machine (Non-linear SVM)
- K-Nearest Neighbor (KNN)
- Logistic Regression
- Decision Tree classification
- Random Forest classification
- Naive Bayes classifier
- XGBoost
- LightGBM & many more...

Algorithms for each type of Machine Learning (Cont...)

Supervised Machine Learning - Regression

- Simple Linear Regression
- Multiple Linear Regression
- Polynomial Regression
- Support Vector Regression
- Decision Tree Regression
- Random Forest Regression
- Lasso Regression
- Ridge Regression
- Elastic Net Regression & many more...

Algorithms for each type of Machine Learning (Cont...)

Unsupervised Machine Learning - Clustering

- K-Means clustering
- Hierarchical clustering & many more...

Unsupervised Machine Learning - Association Rule Learning

- Apriori
- Eclat & many more...

Unsupervised Machine Learning - Dimensionality Reduction

- Principal Component Analysis(PCA)
- Linear Discriminant Analysis(LDA)
- Kernel PCA & many more...

Reinforcement Learning

- Upper Confidence Bound
- Thompson Sampling

Preferable languages used for Machine Learning

Table: Tug of war between languages

Python	**R**	**Julia**
General purpose	Statistical analysis	Scientific computing
Good	Good	**speed** & **performance**
Huge community	Huge community	small community
200k libraries	15k libraries	3k libraries
In Billions	In Billions	13M downloads
-	-	**Compile just in time**
Jupyter, Pycharm	R Studio	Juno IDE
ijulia	-	-

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

So, Why Python ?

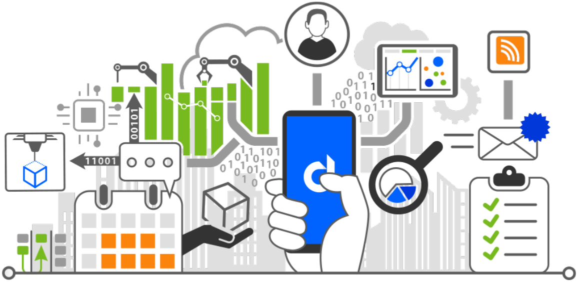
- Open source
- Platform Independence
- User Friendly and Easy to learn
- Vast community support
- Good Visualization options
- A great library ecosystem like Pandas, Scikit.learn, Numpy, Scipy, Keras, Tensorflow, Matplotlib, NLTK, Scikit-image, PyBrain, Caffe, StatsModels
- Capability of interacting with almost all the third-party languages and platforms.

My personal favourites - IDEs

- Anaconda (Distribution) - Jupyter notebook, Spyder
- Colaboratory - Google
- Thonny
- Visual Studio Code

Step 0: Problem definition

Step 1: Data Collection



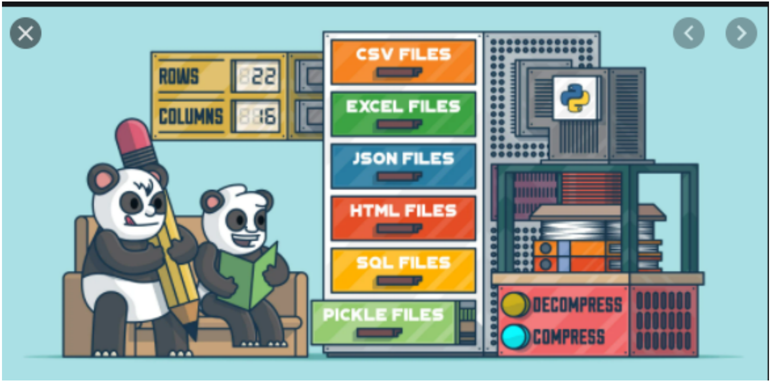
- By using Sensors, Medical devices like ECG, PPG...
- Google dataset search - link
- UCI Machine Learning Repository - link
- CMU libraries - link
- OpenML - link
- Fivethirtyeight - link
- Physionet - link
- Kaggle datasets - link
- Data.gov - link
- Academic torrents - link
- Awesome dataset by github - link

Step 2: Importing libraries

- Either installing or importing pacakages
- Through PIP install! - Through conda install

Step 3: Loading Datasets

- .csv, .json, .xlsx, .xml, .docx, .txt, .pdf, .png, .jpg, .mp3, .mp4



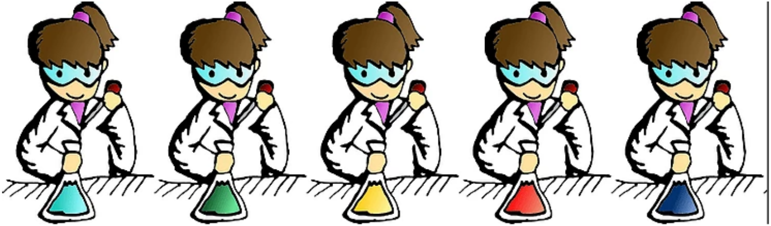
Step 4: Pre-Processing & Exploratory Data Analysis

- 60 - 70 work in this step
- Data information (.head, .tail, .shape, .columns)
- Overall data type (.info)
- Understanding basic statistics of data (.describe)
- Target details (.unique, .valuecounts)
- Checking for missing values (.isnull.sum)
- Solution for missing values (SimpleImputer)
- Outlier detection
- Skewness (log transformation, square root transformation, box-cox transformation) and Kurtosis of data
- Correlation between features (.corr)
- Dependent and Independent variables
- Encoding categorical data for both dependent and independent variables (label encoder, OneHotEncoder, pd.getdummies)

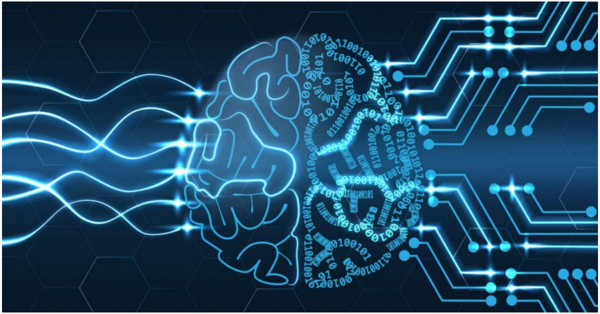
Step 5: Splitting of datasets

Step 6: Feature Engineering

Step 7: Modelling



Step 8: Model Training

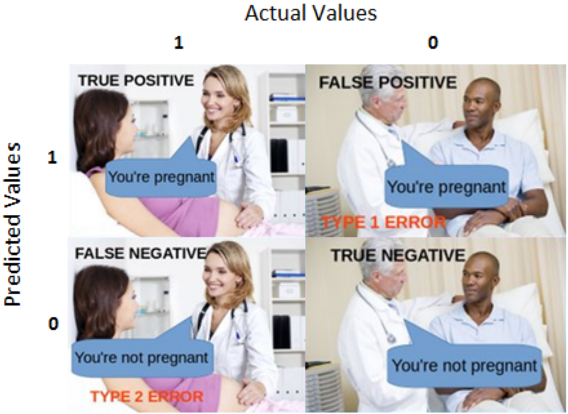


Step 9: Metrics

Metrics - Regression

S No	Term	Criterion
1	R-Squared	High
2	Adj R-squared	High
3	F-Statistics	High
4	Std.Error	Close to zero
5	t-Statistics	>1.96 <0.05
6	AIC (Akaike Info Crit)	Low
7	BIC (Bayesian)	Low
8	Mallows cp	Should be close to no of target
9	MAPE (Mean Abs Per Err)	Low
10	MSE (Mean Squ Err)	Low
11	MPE (Mean Per Err)	Low
12	Min-Max Acc	High

Metrics - Classification - Confusion matrix 1



Metrics - Classification - Confusion matrix 2

- f1 score $= 2*((precision*recall)/(precision+recall))$

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

$$Accuracy = \frac{TrueNegatives + TruePositive}{TruePositive + FalsePositive + TrueNegative + FalseNegative}$$

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

$$\text{Precision} = \frac{\textit{True Positive}}{\textit{True Positive+False Positive}}$$

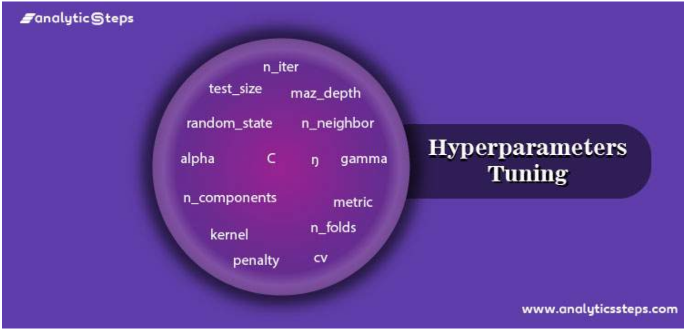
	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

$$\text{Recall} = \frac{\textit{True Positive}}{\textit{True Positive+False Negative}}$$

	Predicted 0	Predicted 1
Actual 0	TN	FP
Actual 1	FN	TP

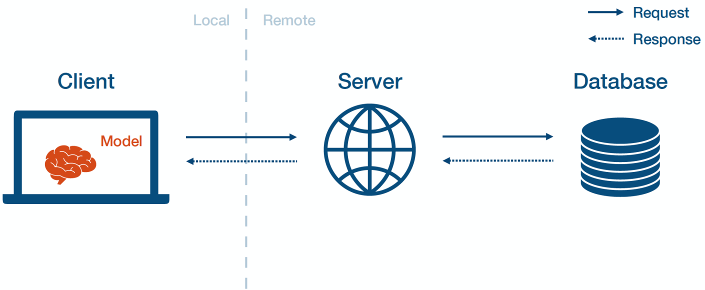
$$\text{Specificity} = \frac{\text{True Negatives}}{\text{True Negatives + False Positives}}$$

Step 10: Hyperparameter tuning

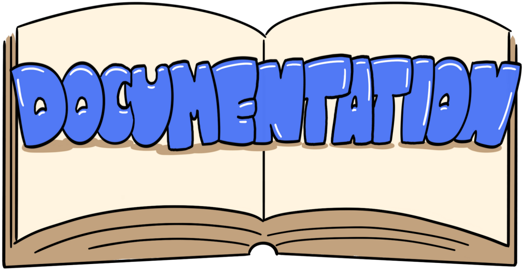


Step 11: Model Validation

Step 12: Model Deployment



Step 13: Documentation



Machine Learning vs. Deep Learning (Part 1)

Aspect	**Machine Learning**	**Deep Learning**
Architecture	Uses various algorithms and models, often simpler	Utilizes deep neural networks with many layers for complex tasks
Feature Engineering	Manual feature engineering is common	Automatically learns and extracts features
Performance	Effective for various tasks, may not excel in very large datasets	Excels in tasks with massive datasets and complexity

Machine Learning vs. Deep Learning (Part 2)

Aspect	Machine Learning	Deep Learning
Computational Requirements	Generally less computationally intensive	Requires powerful hardware (e.g., GPUs/TPUs)
Interpretability	Often more interpretable due to simpler models	Less interpretable, considered "black boxes"
Example Tasks	Spam detection, regression, decision trees	Image recognition, speech recognition, NLP

Deep Learning

- Inspired by the structure and function of the human brain
- BlackBox
- More data
- Deep Learning - Animation @Simplilearn

Types of Deep Learning

- Feed-forward Neural Network or Multi-layer perceptron
- Convolution Neural Network
- Recurrent Neural Network
- Long Short Term Memory Network (LSTM)
- Gated Recurrent Unit (GRU) Networks
- Transformers
- Autoencoders
- Generative Adversarial Networks (GANs)
- Attention Mechanism and many more...

Neurons (Nodes)

- Neurons are the fundamental building blocks of deep learning models. Each neuron processes input data and produces an output. These neurons are organized into layers.

Layers

- Deep learning models consist of multiple layers of neurons, typically arranged in a sequential fashion.
- The input layer receives data, hidden layers process it, and the output layer produces the final result.
- Common layer types include input, hidden (including convolutional and recurrent layers), and output layers.

Weights and Biases

- Weights and biases are parameters associated with each connection between neurons.
- Weights determine the strength of connections and are adjusted during training to learn patterns in data.
- Biases help neurons capture patterns that may not be apparent from the raw data.

Activation Functions

- Activation functions introduce non-linearity into the neural network, allowing it to model complex relationships.
- Common activation functions include ReLU (Rectified Linear Unit), sigmoid, and tanh.

Loss Function (Cost Function)

- The loss function quantifies how well the model's predictions match the actual target values.
- The goal during training is to minimize the loss function by adjusting weights and biases.

Optimization Algorithm

- Optimization algorithms like stochastic gradient descent (SGD) are used to update the model's weights and biases in a way that minimizes the loss function.
- Variants of SGD, such as Adam and RMSprop, are commonly used.

CNN - Components

- Input layer
- Convolutional layers
- Activation layer
- Pooling (Subsampling) layer
- Fully Connected (Dense) Layers
- Flattening Layer
- Output Layer
- Dropout and Regularization
- Normalization Layers (Batch Normalization)
- Padding
- Strides
- Skip Connections (Residual Connections)

Input Layer

- The input layer receives the raw data, typically in the form of images or grids of data (e.g., pixel values in an image).

Convolutional Layers

- Convolutional layers are the core building blocks of CNNs. They consist of multiple filters (also called kernels) that slide over the input data to extract local features.
- Each filter captures specific patterns or features, such as edges, corners, or textures.
- Convolution operations involve element-wise multiplications and summations between the filter and a region of the input, producing feature maps.

Activation Function (ReLU)

- After each convolution operation, a Rectified Linear Unit (ReLU) activation function is applied element-wise to introduce non-linearity.
- ReLU helps the network learn complex and non-linear patterns in the data.

Pooling (Subsampling) Layers

- Pooling layers are used to downsample feature maps and reduce their spatial dimensions.
- Common pooling methods include max-pooling and average-pooling, which retain the most significant information in the feature maps while reducing computational complexity.

Fully Connected (Dense) Layers

- Fully connected layers are traditional neural network layers in which every neuron is connected to every neuron in the previous and subsequent layers.
- These layers enable high-level feature combinations and are typically used in the later stages of a CNN.

Flattening Layer

- Before connecting the convolutional layers to the fully connected layers, the feature maps are flattened into a one-dimensional vector.

Output Layer

- The output layer produces the final predictions or classifications based on the learned features.
- The activation function in the output layer depends on the task; for example, softmax is commonly used for multi-class classification.

Dropout and Regularization

- Dropout layers may be added to mitigate overfitting by randomly deactivating a fraction of neurons during training.
- Regularization techniques such as L1 or L2 regularization can also be applied to the fully connected layers.

Normalization Layers (Batch Normalization)

- Batch normalization layers help stabilize training by normalizing the inputs to each layer.
- They reduce internal covariate shift and improve the convergence of the network.

Padding

- Padding is sometimes added to the input data to control the spatial dimensions of feature maps after convolution.
- Zero-padding is a common technique used to maintain spatial information.

Strides

- Strides determine how much the filter moves across the input data during convolution.
- Strides affect the spatial resolution of feature maps

Skip Connections (Residual Connections)

- Skip connections allow information to bypass certain layers, promoting the flow of gradients during training.
- Residual connections are commonly used to build deeper networks while mitigating the vanishing gradient problem.

Variants of CNN architecture

- LeNet-5
- AlexNet
- VGGNet (VGG)
- GoogLeNet (Inception)
- ResNet (Residual Network)
- MobileNet
- DenseNet (Densely Connected Convolutional Networks)
- EfficientNet
- YOLO (You Only Look Once)
- UNet
- Attention-Based Models (e.g., Vision Transformers) and many more...

VIT Vellore Institute of Technology (Deemed to be University under section 3 of UGC Act, 1956)

Application of CNN architecture

- Image Classification
- Object detection
- Image Segmentation
- Face Recognition
- Gesture Recognition
- Emotion detection
- Medical Imaging
- Video analysis
- Art restoration
- Self-driving cars
- Document analysis and many more...

Some best course for language and technology across the globe

- Python (Dr. Charles Severance)
- 100 Days of Code: The Complete Python Pro Bootcamp for 2023 - Udemy
- Udemy - Machine Learning A-Z (Python and R in Data Science)
- Machine Learning Mastery - Machine Learning track
- Datacamp - Machine Learning Scientist with Python - Free subscription!

Tips to improve ourself & marketing US!

- Open GitHub repository & and start coding from scratch for different dataset Github link - Details about Github - Hit me!
- Try to participate, competitions in Kaggle website for major attractions. Eg: Abhishek Thakur (Approaching (Almost) Any Machine Learning Problem) - Kaggle link
- Try to follow worlds top scientist, R & D, some reowned personalities in the field of Data science, Machine Learning and Deep Learning for their work and tips - Linkedin link

mail me: er.anandprem@gmail.com / premanand.s@vit.ac.in
ring me: +91 73586 79961
follow me: Linkedin
website: anandsdata
author at Analytics Vidhya: premanand17
instagram: premsanand

Learning gives Creativity, Creativity leads to Thinking, Thinking provides Knowledge, and Knowledge makes you Great - Dr APJ Abdul Kalam