PROGRAMMING CLUB, IITK

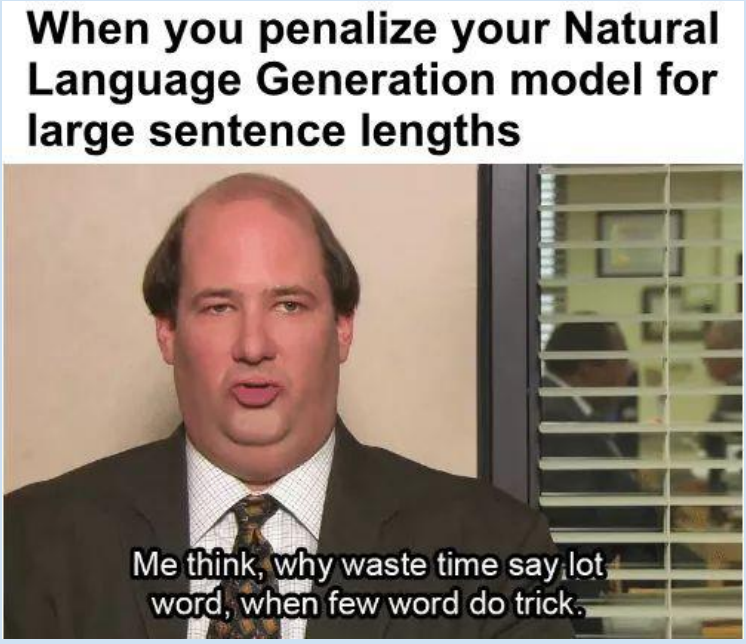

# Roadmap to
# Machine Learning (ML)

---

Welcome to the next series in Tech Trek where we will be giving you a holistic and hands-on introduction to the world of Machine Learning. The roadmap is divided into 6 weeks of content and each week progressively builds upon a new concept. The roadmap includes a lot of material and assignments, which while not trivial are definitely doable given the right amount of enthu and determination.

Also do remember, in case you face any issues, the coordies and secies @Pclub are happy to help. But a thorough research among the resources provided before asking your queries is expected :)

We hope that by the end of the roadmap you are able to understand memes like:


# Week 1 (Reviewing Python, introduction to numpy, pandas and matplotlib)

## Day 1 - Get to know a bit about Machine Learning (A literature survey)


*You sure will be able to make better ML algos after tech trek!*

One must have a broad understanding of what the subject is at hand. Machine learning is a wide field with various domains. It would be really helpful if one goes through a couple of YouTube videos and/or blogs to get a brief hang of it, and its importance. This video by TedEd is a must watch. This blog is also an interesting read. It also cover the different types of machine learning algorithms you will face.

Another fun way to utilize ML:- the science behind lofi music

## Day 2 & Day 3 - Learn a programming language, duh!

There are many programming languages out there, of which only two are suitable for ML, namely Python and R. We recommend any beginner start with Python. Why?

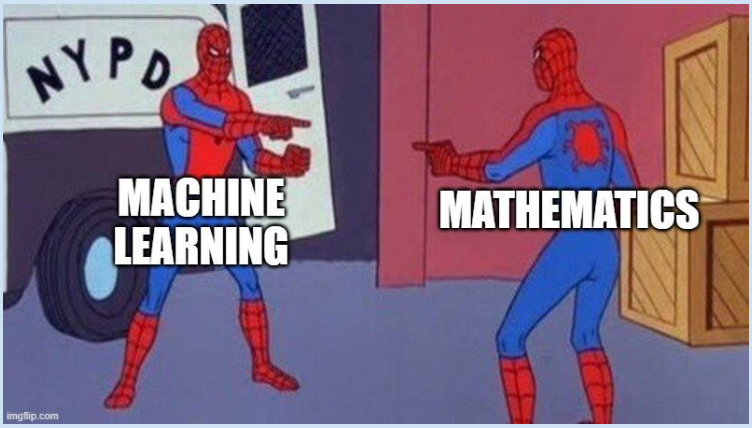
For one, it provides a vast selection of libraries, namely NumPy, pandas, sklearn, TensorFlow, PyTorch, etc., which are super helpful and require little effort for Machine Learning and data science.
Before starting, it is important to setup python in your device, using this as a reference.

Learning python is not hard. Here are a few resources which will teach you about the language swiftly:-

- Medium Blog
- YouTube Video by Free Code Camp

In case you come across a weird syntax or want to find a solution to a problem, the official documentation is the best way to resolve the issues!

### Day 4 - Start to get a hang of some of the inbuilt libraries like NumPy




Mathematics is the heart of Machine Learning. You will get a taste of this statement from *Week 2*. Implementing various ML models, loss functions, and confusion matrix need math.
Mathematics is thus the foundation of machine learning. Most of the mathematical tasks can be performed using NumPy.

The best way to learn about libraries is via their official documentation.
Other resources are as follows:-

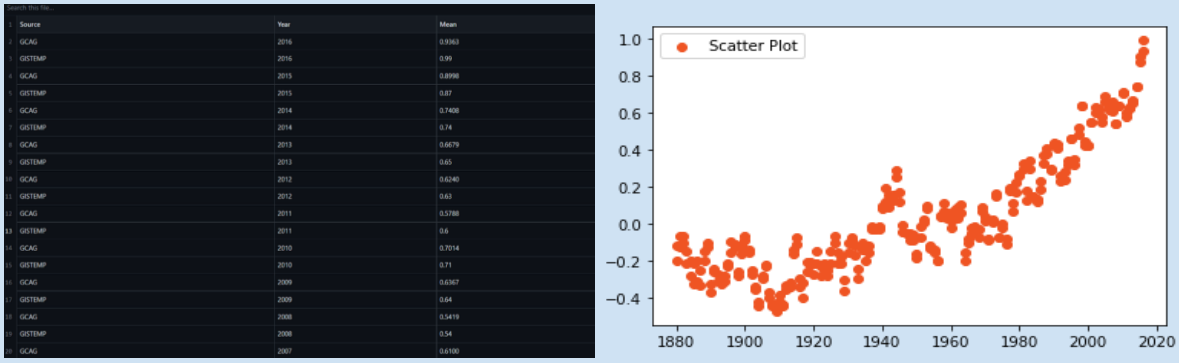
- Video By Free Code Camp
- Numpy in 15 minutes

### Day 5 - Proceed by exploring the other library, Pandas

Data is what drives machine learning. Analyzing, visualizing, and cleaning information is an essential step in the process. For this purpose, Panadas comes to the rescue!
Pandas is an open-source python package built on top of Numpy and developed by Wes McKinney.
Like NumPy, Pandas has official documentation, which you may refer to here.
Other resources are as follows:-

- Medium Blog by Paritosh Mahto
- Pandas in 15 minutes

### Day 6 - Matplotlib - a powerful tool for visualization:




*The average annual temperature above the industrial era around the globe*

Both of the above figures show the same data. However, it is easier to visualize and observe patterns in the second image. (A scatter plot)

Matlotlib is a powerful library that provides tools (histograms, scatter plots, pie charts, and much more) to make sense of data.
The best source to refer to is the documentation in case of discrepancies.
Below are the links to some valuable resources covering the basics of Matplotlib:-

- Code With Harry
- Free Code Camp

**Day 7 - Play around in Kaggle**

Use this day as a practice field, to utilize all your skills you learnt. Head over to Kaggle and download any dataset you like. Apply the skills you procured and analyze trends in different data sets. Here is a brief walkthrough of the UI.
All about Kaggle

---

# Week 2 (Basic Mathematics for ML, KNN, Linear Regression)

## Day 1 - Descriptive Statistics

By now, you must have been comfortable with processing data using python libraries. Before going further, let us recall some basic concepts of Maths and Statistics. Follow these resources:

- Mean, Variance, Standard Deviation - Read theory from the book, Statistics, 11th Edition by Robert S. Witte, sections 4.1 - 4.6.
- Gaussian Distribution - Read this blog.
- Correlation in Variables - Read theory from the book, Statistics, 11th Edition by Robert S. Witte, Chapter 6.

## Day 2 - Regression Problems

The common problems you would attempt to solve using supervised machine learning can be categorised into either a regression or a classification one. For e.g., Predicting the price of a house is a regression problem while classifying an image as a dog or cat is a classification problem. As you can clearly see, when you are outputting a value (real number), it is a regression problem, while predicting a category (class), we can call it a classification problem.

Go through this article on linear regression. Now for a deeper intuition watch this video. Now follow this article to understand the implementation using various libraries. If you are further interested, you may see Statistics, 11th Edition by Robert S. Witte, Chapter 7.

## Day 3 - Classification using KNNs

KNNs are one of the first classification algorithms. Watch first 5 videos of this playlist to know more. Try to implement a KNN on your own following this tutorial.

### Day 4 - Probability and Inferential Statistics

Knowledge of probability is always useful in ML Algorithms. It might sound a bit of an overkill, but for the next two days we will revise some concepts in probability. You can use your JEE Notes or cover theory from the book, Statistics, 11th Edition by Robert S. Witte, sections 8.7 - 8.10. Go through important theorems like Baye's Theorem and Conditional Probability. Audit the Coursera Inferential Statistics Course for free and complete Week 1 upto CLT and Sampling.

### Day 5 - Inferential Statistics Continued

Complete remaining portion of Week 1 and Week 2 of the Inferential Statistics course. You can also use the book, Statistics, 11th Edition by Robert S. Witte, as a reference.

### Day 6 - Naïve Bayes Classifier (Both Multinomial and Gaussian)

These are another kind of classifiers, which just work on Bayes Theorem. For Multinomial Naïve Bayes Classifier, watch this. Then watch the corresponding Gaussian Classifier here. If interested, you can also go through this Wikipedia article.

### Day 7 - Preparation for Next Week

Towards the end of the week, let us revise some tools in linear algebra. This has some motivation regarding the content. Revise Vectors, Dot Product, Outer Product of Matrices, Eigenvectors from MTH102 course lectures here. Revise some concepts on multivariable mathematics (MTH101) here.

---

# Week 3 (Logistic Regression, Decision Trees)

By now you should have a grasp over what regression means. We have seen that we can use linear regression to predict a continuous variable like house pricing. But what if I want to predict a variable that takes on a yes or no value. For example if I give the model an email, can it tell me whether it is spam or not? Such problems are called classification problems and they have very widespread applications from cancer detection to forensics.

This week we will be going over two introductory classification models: **logistic regression** and **decision trees.**

## Day 1 - Review of Cost functions, hypothesis functions and gradient descent.

As you may recall, the **hypothesis function** was the function that our model is supposed to learn by looking at the training data. Once the model is trained, we are going to feed unseen data into hypothesis function and it is magically going to predict a correct (or nearly correct) answer!

The hypothesis function is itself a function of the **weights** of the model. These are parameters associated with the input features that we can tweak to get the hypothesis closer to the ground truth.

But how do we ascertain whether our hypothesis function is good enough? That's the job of the **cost function**. It gives us a measure of how poor or how wrong the hypothesis function is performing in comparison to the ground truth.

Here are some commonly used cost functions: https://www.javatpoint.com/cost-function-in-machine-learning

**Gradient descent** is simply an algorithm that optimizes the weights of the hypothesis function to minimize the cost function (i.e to get closer to the actual output).

[For now you may treat gradient descent as a black box, but the algorithm itself has some mathematical intricacies. You may refer to https://towardsdatascience.com/gradient-descent-algorithm-a-deep-dive-cf04e8115f21
for a clearer understanding of its mechanism]

For a refresher on what these terms were and how they pertained to linear regression have a look at this article: https://medium.datadriveninvestor.com/linear-regression-hypothesis-function-cost-function-and-gradient-descent-part-1-6cd865552923

## Day 2 - Logistic Regression

The logistic regression model is built similarly to the linear regression model, except that now instead of predicting values in a continuous range, we need to output in binary i.e 0 and 1.

Let's take a step back and try to figure out what our hypothesis should be like. A reasonable claim to make is that our hypothesis is basically the probability that y=1 (Here y is the label i.e the variable we are trying to predict). If our hypothesis function outputs a value closer to 1, say 0.85 it means it is reasonably confident that y should be 1. On the other hand if the hypothesis outputs 0.13 it means there is a very low probability that y is 1, which means it is probable that y is 0. Now, building upon the concepts from linear regression, how do we restrict (or in the immortal words of 3B1B - "squishify") the hypothesis function between 0 and 1? We feed the output from the hypothesis function of the linear regression problem

into another function that has a domain of all real numbers but has a range of (0,1). An ideal function with this property is the **logistic function** which looks like this: https://www.desmos.com/calculator/se75xbindy. Hence the name logistic regression.

However, we aren't done yet. As you may remember we used the mean squared error as a cost function. However, if we were to use this cost function with our logistic hypothesis function, we run into a mathematical wall. This is because the resulting hypothesis as a function of its weights would become non-convex and gradient descent does not guarantee a solution for non-convex functions. Hence we will have to use a different cost function called binary cross-entropy.
Have a look at these articles:

- https://bit.ly/3FAxwI5
- https://towardsdatascience.com/understanding-binary-cross-entropy-log-loss-a-visual-explanation-a3ac6025181a

We can now use gradient descent.

For a thorough understanding of how to use the above concepts to build up the logistic regression model refer to this article:
https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc

**Day 3 & 4 - Getting your hands dirty with code**

One thing you must always keep in mind is that while learning about new concepts, there is no substitute for actually implementing what you have learnt.

Spend this day trying to come up with your own implementation for logistic regression. You may refer to other people's implementations.
This will surely not be an easy task but even if you fail, you will have learnt a lot of new things along the way and have gotten a glimpse into the inner workings of Machine Learning.
Some resources to help you get started:

- https://github.com/SSaishruthi/LogisticRegression_Vectorized_Implementation/blob/master/Logistic_Regression.ipynb
- https://towardsdatascience.com/building-a-logistic-regression-in-python-step-by-step-becd4d56c9c8
- https://realpython.com/logistic-regression-python/

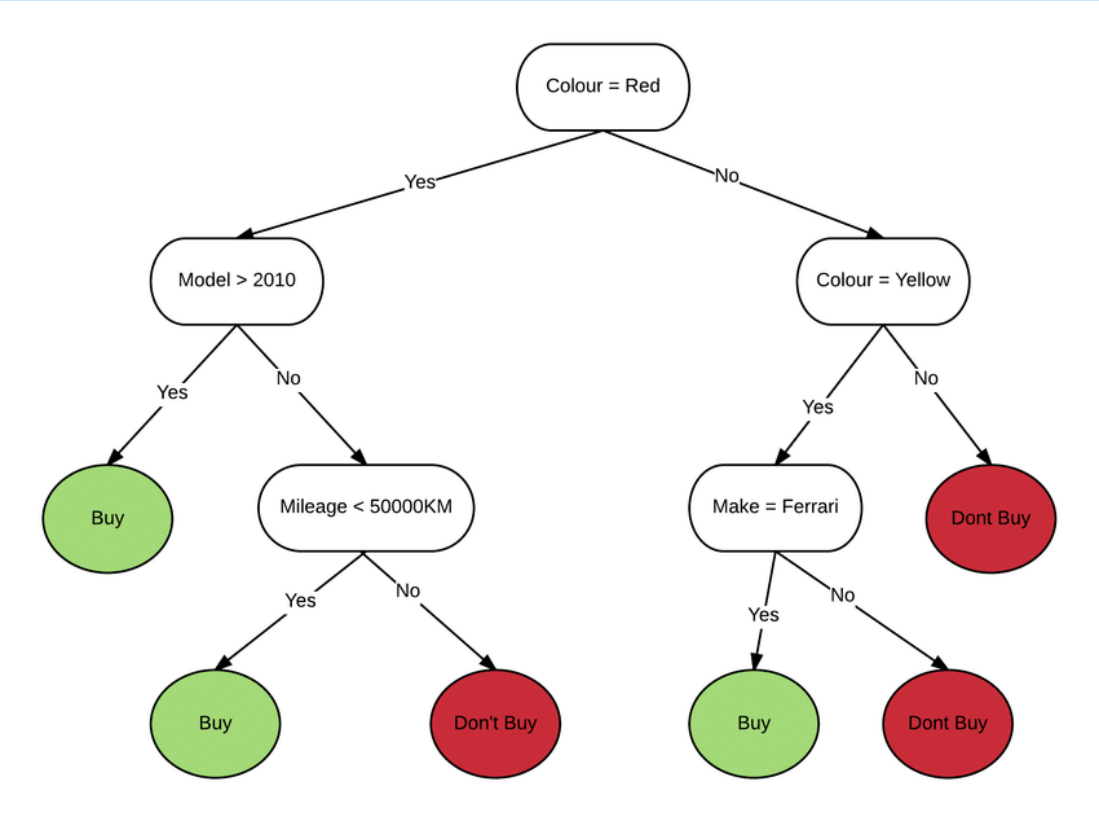
Some datasets you might want to use to train your model on:

- Iris Dataset - https://archive.ics.uci.edu/ml/datasets/iris
- Titanic Dataset - https://www.kaggle.com/c/titanic/data
- Bank Marketing Dataset - https://archive.ics.uci.edu/ml/datasets/bank+marketing#
- Wine Quality Dataset - https://archive.ics.uci.edu/ml/datasets/wine+quality

## Day 5 & 6 - Decision Trees

Now let's move onto another interesting classification paradigm: the decision tree.
This model is even more powerful and intuitive than logistic regression. You have probably used a decision tree unconsciously at some point in your life to take a decision. There is even an example of it on the IITK Freshers Intro 2021 [brownie points for finding it]!

In case you aren't familiar with what a tree is have a look at https://www.programiz.com/dsa/trees




Above is a decision tree someone might use to decide whether or not to buy a specific car.

Let's have a look at a few mathematical considerations before developing further

- https://towardsdatascience.com/understanding-entropy-the-golden-measurement-of-machine-learning-4ea97c663dc3
- https://www.analyticsvidhya.com/blog/2020/11/entropy-a-key-concept-for-all-data-science-beginners/
- https://medium.com/analytics-steps/understanding-the-gini-index-and-information-gain-in-decision-trees-ab4720518ba8

Moving on to how to build a decision tree, refer to the following:

- https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052
- https://www.ibm.com/in-en/topics/decision-trees

For a more visual understanding you can refer to the following video:
https://youtu.be/ZVR2Way4nwQ

**Day 7 - Implementing a Decision Tree**

So far we have not really looked into much code. But as always no concept is complete without implementation.
Your task for today will be to implement a decision tree.
You may refer to following for help:

- https://machinelearningmastery.com/implement-decision-tree-algorithm-scratch-python/
- https://towardsdatascience.com/implementing-a-decision-tree-from-scratch-f5358ff9c4bb
- https://www.kaggle.com/code/prashant111/decision-tree-classifier-tutorial
- https://towardsdatascience.com/an-exhaustive-guide-to-classification-using-decision-trees-8d472e77223f

Finally, solve the Titanic challenge on Kaggle:
https://www.kaggle.com/competitions/titanic

---

## Week 4 (Perceptrons, Neural Networks, and an introduction to Tensorflow)

The last 3 weeks of this roadmap will be devoted to Neural Networks and their applications.

### Day 1 - The perceptron and Neural Networks

The ultimate goal of Artificial Intelligence is to mimic the way the human brain works. The brain uses neurons and firing patterns between them to recognize complex relationships between objects. We aim to simulate just that. The algorithm for implementing such an "artificial brain" is called a neural network or an artificial neural network (ANN).
At its core, an ANN is a system of classifier units which work together in multiple layers to learn complex decision boundaries.

The basic building block of a neural net is the **perceptron.** These are the "neurons" of our neural network.
https://towardsdatascience.com/what-is-a-perceptron-basics-of-neural-networks-c4cfea20c590
Do you see the similarity with logistic regression?

Moving on to how we build up a neural network from perceptrons.
https://medium.com/deep-learning-demystified/introduction-to-neural-networks-part-1-e13f132c6d7e
https://medium.com/ravenprotocol/everything-you-need-to-know-about-neural-networks-6fcc7a15cb4

You should also watch the first video in 3Blue1Brown's series on Neural Networks.
https://www.youtube.com/watch?v=aircAruvnKk&list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

## Day 2 - Backpropagation

Now you might be wondering how to train a neural network. We are still going to be using gradient descent (and its optimized forms) but to calculate the gradients of the weights we will be using a method called backpropagation.
https://machinelearningmastery.com/difference-between-backpropagation-and-stochastic-gradient-descent/

To get a visual understanding you are encouraged to watch the remaining 3 videos of 3B1B's series on Neural Networks.
https://www.youtube.com/playlist?list=PLZHQObOWTQDNU6R1_67000Dx_ZCJB-3pi

Refer to the following articles get a feel of the mathematics under the hood.
https://towardsdatascience.com/deriving-the-backpropagation-equations-from-scratch-part-1-343b300c585a
https://towardsdatascience.com/understanding-backpropagation-algorithm-7bb3aa2f95fd

## Day 3 & 4 - Implementation

Dedicate these two days to try to implement a neural network in python from scratch. Try to have one input, one hidden and one output layer.

You may train your model on any of the datasets given in week 3.

https://towardsdatascience.com/an-introduction-to-neural-networks-with-implementation-from-scratch-using-python-da4b6a45c05b

**Day 5 - Tensorflow**

We will now start building models. For this, we are going to use Tensorflow. It enables us to implement various models that you have learnt about in previous weeks.
The TensorFlow platform helps you implement best practices for data automation, model tracking, performance monitoring, and model retraining.
First, install tensorflow. Follow this link:
https://www.tensorflow.org/install
Keras - Keras is the high-level API of TensorFlow 2: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity.
https://www.tutorialspoint.com/keras/keras_installation.htm
Go through this article to know more about keras and tensorflow.
https://towardsdatascience.com/tensorflow-vs-keras-d51f2d68fdfc
Through the following link, you can access all the models implemented in keras, and the code you need to write to access them
https://keras.io/api/

You can have a brief overview of the various features keras provides.
https://www.tensorflow.org/tutorials
Go through the above tutorial. There are various subsections for keras basics, loading data and so on. These will give you an idea on how to use keras and also how to build a model, process data and so on. You can see Distributed Training section if you have time, but do go through other sections.

**Day 6 & 7**

You will now do a project. You can make use of models available in keras. You would also need to use some pre-processing.
Go to this link, download the dataset and get working!
https://www.kaggle.com/datasets/mirichoi0218/insurance
P.S. some features aren't as useful as others, so you may want to use feature selection
https://www.javatpoint.com/feature-selection-techniques-in-machine-learning.
Don't worry if your results aren't that good, this isn't really a task to be done in 1 day. It's basically to give you a hands on experience. Also, there are notebooks available on kaggle for this problem given, you can take hints from there as well. There are many similar datasets available on kaggle, you can try those out too! Also, as you learn more topics in ML, you will get to know how to further improve accuracy. So just dive in and make a (working) model.
Some More Topics :
Following are some links which cover various techniques in ML. You may not need them all in this project, but you may need them in future. Feel free to read and learn about them anytime, they are basically to help you build more efficient

models and process the data more efficiently. Thus, you can cover these topics in future as well.

https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/

https://towardsdatascience.com/regularization-in-machine-learning-76441ddcf99a

https://towardsdatascience.com/train-validation-and-test-sets-72cb40cba9e7

https://machinelearningmastery.com/k-fold-cross-validation/

---

# Week 5 (Convolutional Neural Networks)

## Day 1

Now having understood the basics of logistic regression, tensorflow and image processing, its time to further our understanding and move into the domain of Computer Vision. Computer Vision is all about giving 1s and 0s the ability to see. A Convolutional Neural Network is a Deep Learning algorithm that can take in input images and be able to differentiate one from the other or maybe identify some objects in the image.

Convolution is a mathematical operation to merge two sets of information. Convolution is covered quite extensively in the last few lectures of MTH102. However, you don't require any of that knowledge to understand the concept here. (Although knowing it will give you a better insight in the algorithm)

To get a basic understanding of what CNN is about, this blog is a very good read. Introduction to CNN (Read Here).

If you are more comfortable reading in a research based format, this paper is also a good read. Read Here.

## Day 2

Now its time to revise and implement some part of what we did yesterday and implement it in one of the applications.

Its best to cover the week 1 of this course by Andrew NG on coursera. You can audit the course for now. Week 1 CNN Course. The videos are more or less the same as what you had read in the blog.

The assignments of the course are available on this GitHub repository. You can clone the whole repository and open the C4 convolutional neural networks folder followed by week 1. You'll find all the programming assignments here.
GitHub Repository. Please check out the issues section of this repository as well.

### Day 3

This completes our basic understanding of CNNs. Now, Lets understand some basic networks that are required to implement CNNs. In this roadmap, for the week, we'll tell you to cover Residual Networks only, however to get an in depth knowledge of the topic, you can read about the other networks (MobileNet, etc) later on.
For day 3, Watch the week 2 lectures on ResNet (First 5 videos of week 2) of that course on coursera and solve the first assignment on residual networks (using the same GitHub repository).

### Day 4

Watch the remaining portion of the week 2 lectures from the course and solve the assignment on Transfer Learning using MobileNet.
To get a more in depth knowledge of MobileNet you can check out this paper
Read Here

## Day 5

Watch the first 9 lectures of week 3 (until the YOLO detection algorithm) and solve the assignment on car detection with YOLO.
A nice blog on the working of YOLO is available here. You can refer to it alongside the course. Read Here
This blog gives a version-wise overview of the YOLO algorithm models is a good read to get more in depth on how the algorithm was developed and what is next in this domain. Read Here

## Day 6

Cover the remaining portion of week 3 and try to do the assignment on image segmentation with U net from the GitHub repository.

## Day 7

From the week 4 of the course watch the first 5 videos on face recognition and try to do the assignment on face recognition from the GitHub repository. You can skip over the last part on style generation on neural style transfer of the course for now. You can try it out later.
This is a GitHub repository for implementing the facenet model in keras. You can check it out to go more in depth. Click Here.

---

# Week 6 (Introduction to Natural Language Processing)

For the last week, we will cover more advanced and interesting applications of ML. We will also go through models which deal with sequential data. This week will challenge your skills and also introduce you to the kinds of models you need to solve real-time problems.
Natural Language Processing (NLP) is the set of methods for making human language accessible to computers. The goal of NLP is to provide new computational capabilities around human language: for example, holding a conversation, summarizing or making out meaningful inferences from an article, and so on.

I) For a computer to work with any concept it is necessary that there should be a way to express the said concept in the form of a mathematical model.
A digital computer cannot understand words. The first challenge we deal with is to make machines understand the text as numbers.

## Day 1

Wordnet– A lexical database of semantic relationships between words
Wordnets due to many limitations are not feasible in a practical scenario,as you'll see further, in modern NLP words are represented as vectors!
Now, that may seem a bit counter intuitive, read further→ → Word_embeddings
Also get an idea of One Hot Encodings in NLP.

Lets see some common methods used to generate word embeddings which broadly classified into two—

i) **Frequency Based-**

- Bag of Words or CountVectorizer()
- Hashing Vectorizer
- TFIDF or TfidfVectorizer()

One thing that can be easily made out by frequency based methods is that while training your corpus needs to extremely large to achieve effective performance. But what if we had only relatively small dataset? That's where pretrained embeddings shine!

**Day 2**

ii) **Pretrained Embeddings-** Word embeddings can be learned from text data and reused among projects.

- An introduction to Word2Vec Also you might want to get familiar with the working of word2vec.
- GloVe-[Read here]
  Also read this. Since we know now how make our language accessible to computers, let's move on neural nets!
- Models using Deep learning - Get familiar with Sequential Learning [or read this].

**Day 3**

1)Recurrent Neural Networks
By learning rnn your journey of NLP with deep learning truly starts here. RNNs by themselves are of little use, but they form the building blocks of many bigger models.

The key to completely understand rnn is to implement a RNN model from scratch. You may do it by using numpy in python. Refer to this blog for implementation details.

## Day 4

2) Get introduced to LSTM RNNs.
This article is recommended as it provides an in-depth understanding of usually hard-to-understand LSTMs.
3)GRUs
Rnn vs Lstm vs GRU - This paper evaluates and compares the performance of the three models over different datasets.
4)Bi-RNNs
5)Deep RNNs

## Day 5

Here are some topics for you to go through. We will first learn about Sequence to sequence models. Then we will use it as an example to study Attention Mechanism.
6)Seq2Seq
7)Attention-Mechanism

Now, we will cover Encoders and Decoders part of LSTM model. These will help you better understand time series models.
8)Encoder-Decoder, also read this.
9)Autoencoders

## Day 6

10)Transformers
Now coming to the state of the art models-
BERT (encoder only transforms), also read this.
GPT (decoder only transforms)
Read this interesting blog by IBM which describes why BERT and GPT are so powerful as models.

## Day 7

Now, having loads of theory doesn't really mean anything if you don't apply it! Time to do projects, but before that go through some basics we skipped in the beginning which become essential while practicing NLP.

Text Preprocessing -
Text preprocessing involves transforming text into a clean and consistent format that can then be fed into a model for further analysis and learning. Preprocessing leads to better performance.

Linguistic Essentials- Also read this.

In order to get more insights to a NLP problem, and to understand how it differs form other ML problems, it is essential to have some fundamental knowledge in Linguistics. This becomes more important when you need to process a language other than English. Go through the following links to learn about Linguistics.

### Further resources

- https://sgfin.github.io/files/notes/CS229_Lecture_Notes.pdf
- https://www.coursera.org/specializations/deep-learning
- https://www.youtube.com/playlist?list=PLoROMvodv4rMiGQp3WXShtMGgzqpfVfbU
- http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf
- https://drive.google.com/file/d/1hFPxorU1AMDXE_02HBwM0hAfcwsdOCeo/view?usp=sharing

---


**Contributors**

Anwesh Sen Saha | +91 84519 62003

Dhruv Singh | +91 96216 88942

Kartik Kulkarni | +91 91750 09924

Ridin Datta | +91 74390 79526

Talin Gupta | +91 85589 13121

Tejas Ahuja | +91 87007 94886