FinSearch:

Stock trading and Investment using reinforcement and deep learning

Week-1:

https://drive.google.com/file/d/1A2wdiixzUsA1Ud-o1swd7AGhm7wAxk75/view
Drive link for the overall resources of the project(main resources)
https://www.youtube.com/watch?v=FgzM3zpZ55o&list=PLoROMvodv4rOSOPzutgyCTapiGlY2Nd8u
Please follow 1st 3 videos of the playlist to get insight of RL and how policies work.

Week 1: Introduction to Machine Learning and Python

Howdy Fellas!
Welcome to an exciting journey in the realm of Reinforcement Learning🤖. This week we will be learning some basic concepts that may not seem important but are the basic pillars of anything MLeee or AI-kinda. We will begin with some content on Python and move on to understanding what Machine Learning means. Buckle up as we decode the language of machines and explore the fascinating world of intelligent algorithms. Let the journey commence! 🚀🌈

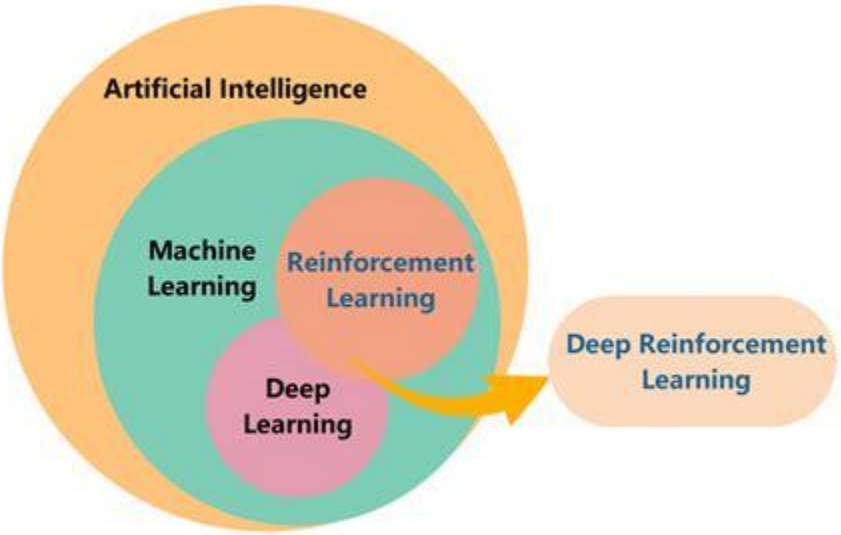
Python

This is some content that will help Noobs master Python really fast and help Masters revise their knowledge (Sadly 😓 Python Syntax knowledge is exponentially decaying). If you already know the content then it is okay to skip some parts.

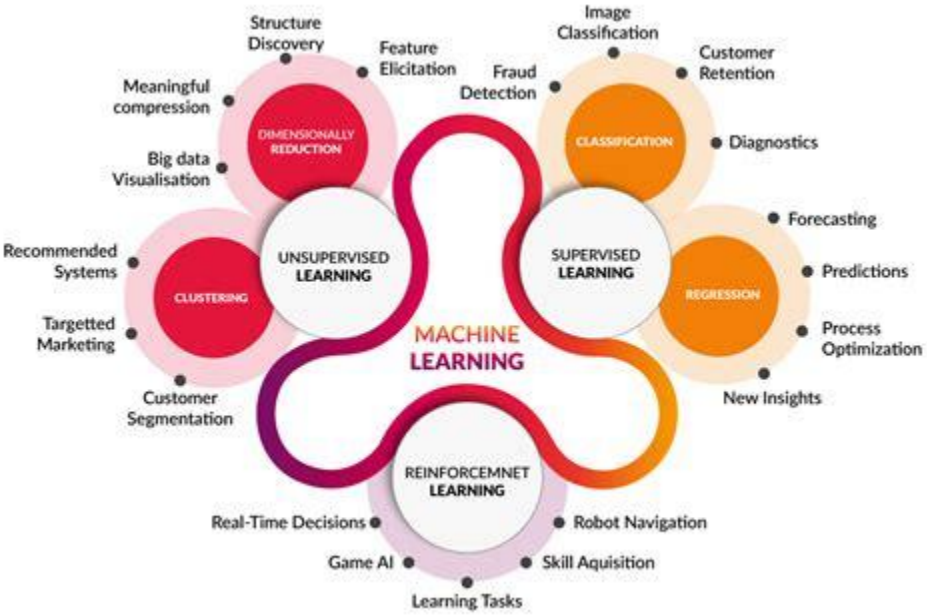
1. Learn Python from Scratch: Learn Data Types, Operators, Conditional Statements, Loops, Strings, Lists, Tuples, Dictionaries, Sets, Functions. (*Optional*: File Handling)
2. Python OOP: Learn Classes & Objects, Inheritance and other concepts

Machine Learning

Machine Learning is the core concept that is present in almost all AI. Reinforcement Learning is a part of Machine Learning though quite different from the traditional approaches previously used.



In this week we are going to learn what Machine Learning means followed by a few simple algorithms used to predict data. We will actually learn what is behind the *black box* of the model and how it works.



We will begin with Supervised Learning and its two major types Regression and Classification.
For the following resources you are required to create an account on Coursera and sign up. While starting the code you DO NOT need to pay anything. Just choose to Audit the course. For help, click here

1. Supervised Learning: Regression & Classification
   Week 1: Introduction to Machine Learning
   - Overview of Machine Learning
   - Supervised vs. Unsupervised Machine Learning
   - Regression Model
   - Train the model with Gradient Descent
2. Supervised Learning: Regression & Classification
   Week 3: Classification
   - Classification with Logistic Regression
   - Cost Function for Logistic Regression
   - Gradient Descent for Logistic Regression
   - Problem of Overfitting

NumPy, Pandas, and Matplotlib

These are simple libraries that have been created to make our work easy. They are highly used and are a must-learn for an ML Specialist. No need to memorize all syntax. You should just be able to use the appropriate functions when required.

1. NumPy
   NumPy is the powerhouse of numerical computing in Python. It provides essential functionality for working with large, multi-dimensional arrays and matrices, along with a collection of high-level mathematical functions to operate on these arrays. NumPy is a fundamental library for scientific computing, enabling efficient and fast operations on numerical data.
   NumPy for Beginners
2. Pandas
   Pandas steps in as the data manipulation maestro, making working with structured data seamless. It introduces data structures like DataFrames, which are akin to tables in a database, allowing for easy indexing, slicing, and reshaping of datasets. Pandas is a game-changer for data preprocessing, cleaning, and analysis in Python, empowering data scientists to handle diverse datasets with ease.
   Pandas for Beginners
3. Matplotlib
   Visualization is key to understanding data, and Matplotlib is the go-to library for

creating static, interactive, and animated plots in Python. Whether you need basic line charts or intricate heatmaps, Matplotlib provides a flexible and customizable platform for data visualization. It complements the analytical power of NumPy and Pandas, turning raw data into insightful visualizations for better comprehension and decision-making.
Matplotlib for Beginners

Extra Resources

1. Supervised Learning: Regression & Classification
   Week 2: Regression with Multiple Input Variables
   - Multiple Linear Regression
   - Gradient Descent in practice
2. NumPy Cheatsheet
3. Pandas Cheatsheet
4. Matplotlib Cheatsheet

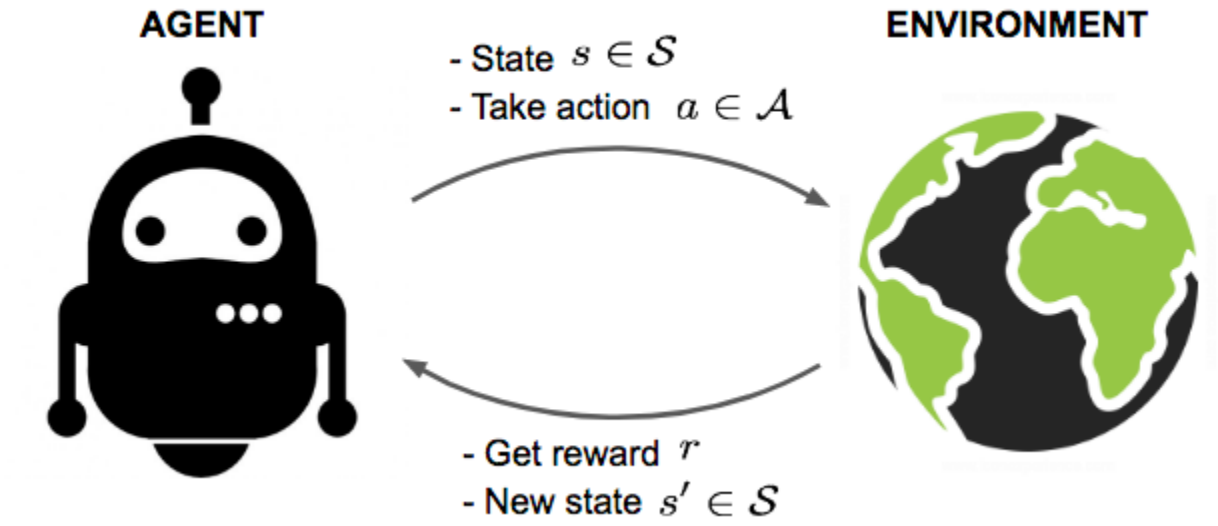
Assignment 1

There will also be an assignment to check your understanding of this week's topic. The deadline will be 11:59 pm, 25th Dec (Mon). The assignment will be training a linear regression model on the given dataset. We will test the model on the hidden test dataset. The assignment will be hosted on Kaggle which is a wonderful platform to test your skills 🚀. This shall give you a brief introduction to this awesome website.
Follow the given steps to access your assignment:

- Create an account on Kaggle
- Join the competition using https://www.kaggle.com/t/54c575b375b34359ad9b8de7613cd65e. ALL THE DETAILS are given in the competition introduction.
- Download the dataset, write the code in Python Notebook(.ipynb file), train the model for the training dataset, and get the predictions for the test dataset.
- Try plotting your model using Matplotlib and see if your model has fitted well to the training dataset.
- Upload the predictions to the Kaggle Competition.
- Enjoy 😎

Week 2: Introduction to Reinforcement Learning

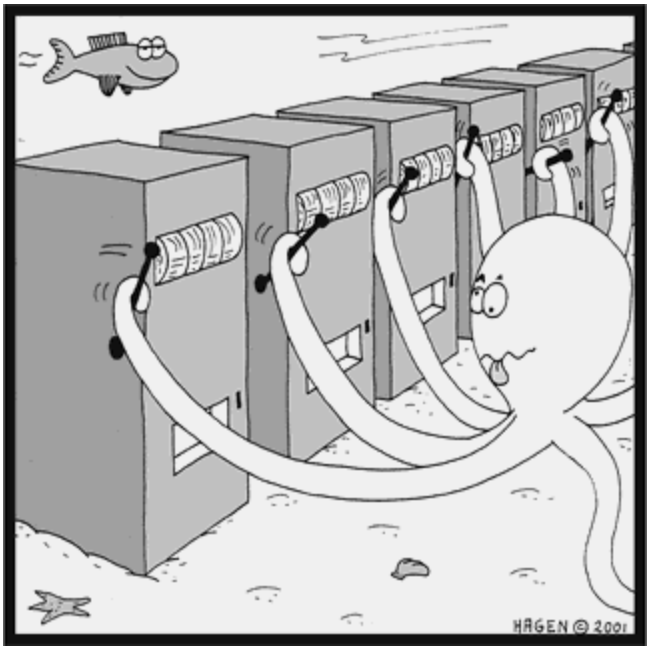
Welcome to Week 2 of our project, where we're diving headfirst into the fascinating world of Reinforcement Learning (RL). Brace yourself for an exhilarating exploration of RL concepts and algorithms that will set the stage for the thrilling challenges that lie ahead.



1. What is Reinforcement Learning - Dive deep into the core principles of RL with this insightful article.
2. A breif introduction to RL (video) - Let's kick things off with an engaging video that brings RL to life.

🎰 Multi-Armed Bandits: RL's Foundation

Now, let's turn our attention to the foundational Multi-Armed Bandit problem. We'll unravel the intricacies of this crucial concept and equip ourselves with three powerful algorithms—Epsilon Greedy, Thompson Sampling, and UCB—to tackle this challenge head-on.



1. Multi Armed Bandit - Explore the fundamentals of Multi-Armed Bandits in this comprehensive resource.
2. Introductory video - Immerse yourself in this video to grasp the essence of Multi-Armed Bandits.

The following are some algoirthms for solving Multi Armed Bandits:

3. Epsilon Greedy Algorithm
4. Thompson Sampling Algorithm
5. Upper Confidence Bound (UCB) Algorithm
6. Thompson vs UCB
7. Formal Notes of Multi Armed Bandits - Take your understanding to the next level with formal notes. The section "Stochastic Multi-Armed Bandits" and its algorithms provide a solid foundation. The optimal bound proof for UCB is not compulsory to read. If you're up for a challenge, explore the optimal bound for UCB in this in-depth explanation.

Assignment 2

Now after some heavy (maybe 🤔?) bombarding of info let's move onto the exciting part. This week you will write your very own algorithm to crack the multi-armed monster bandit. Officially you are supposed to use the problem.ipynb as a base to implement the *Epsilon-Greedy Algorithm, the **Upper Confidence Bound(UCB) Algorithm* and the *Thompson Sampling Algorithm*.
Make sure to make plenty use of classes and objects while implementing the algorithm.
We want you to complete the following tasks:

1. Run the algorithms for some values of the horizon and other parameters (Your Choice Completely).
2. At the end just print the total regret accumulated and what arm do you (your algorithm) thinks is *best*.
3. Also display a graph showing the Total Regret vs Timesteps plot. Judge them and decide which algorithm is the best. (If you can't then at least point out the worst looking one)
4. Remember to do it for all 3 Algorithms
5. For evaluation create a seperate code block where you create a custom bandit (prob. values should be changable by us). Just *running* the code block should run all 3 algorithms for bandits with same probability values and display the plot for all of them.

The *Deadline* will be 11:59 pm, 13st July (Sun).
For submission, you are required to create a GitHub Repository (if you don't have an account make one quickly 🐌). *Upload* the final Notebook(.ipynb) file to the repo. Remember to do the same for Week 1 submissions.
Note that for final evaluation of WIDS, this Repository will be important so make one quickly.