**Short home assignment**

**Name:** Michael Levinger

**1) Please explain the difference between machine learning, deep learning, and AI**

In short, **Artificial intelligence** (AI) is an area of computer science that emphasizes the creation of intelligent machines that work and react like humans.

**Machine Learning** which is an application of AI - is a scientific study of algorithms and statistical models that computer systems use in order to perform a specific task effectively without using explicit instructions, relying on patterns and inference instead. **Deep learning** is part of a broader family of machine learning methods based on artificial neural networks. In more details -

**AI**: In [computer science](https://en.wikipedia.org/wiki/Computer_science), artificial intelligence (AI), sometimes called machine intelligence, is [intelligence](https://en.wikipedia.org/wiki/Intelligence) demonstrated by [machines](https://en.wikipedia.org/wiki/Machine), in contrast to the natural intelligence displayed by humans. Colloquially, the term "artificial intelligence" is used to describe machines/computers that mimic "cognitive" functions that humans associate with other [human minds](https://en.wikipedia.org/wiki/Human_mind), such as "learning" and "problem solving".

[**Machine Learning**](https://www.geeksforgeeks.org/getting-started-machine-learning/): Machine Learning is the learning in which machine can learn by its own without being explicitly programmed. It is an application of AI that provide system the ability to automatically learn and improve from experience. One of the simple definition of the Machine Learning is **“Machine Learning is said to learn from experience E with respect to some class of task T and a performance measure P if learners performance at the task in the class as measured by P improves with experiences*"*.** Machine learning algorithms are often categorized as supervised or unsupervised:

**Supervised** machine learning algorithms can apply what has been learned in the past to new data using labeled examples to predict future events. Starting from the analysis of a known training dataset, the learning algorithm produces an inferred function to make predictions about the output values. The system can provide targets for any new input after enough training. The learning algorithm can also compare its output with the correct, intended output and find errors in order to modify the model accordingly.

In contrast, **unsupervised** machine learning algorithms are used when the information used to train is neither classified nor labeled. Unsupervised learning studies how systems can infer a function to describe a hidden structure from unlabeled data. The system doesn’t figure out the right output, but it explores the data and can draw inferences from datasets to describe hidden structures from unlabeled data.

**Deep Learning:** Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. In deep learning, a computer model can learn to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labeled data and neural network architectures that contain many layers.

**2)  Please choose one of the following algorithms: XGBoost, SVM, Logistic Regression, Random Forest and write the following:**

**A brief description about the algorithm and how it is working -**

I have chosen to write about the Logistic Regression algorithm.

Logistic regression is a classification algorithm used to assign observations to a discrete set of classes (e.g. cat/dog, S/M/L, 0/1). Unlike linear regression which outputs continuous number values, logistic regression transforms its output using the sigmoid function to return a probability value which will lie in the range between 0 and 1.

In order to map the Sigmoid function output to a discrete class (true/false, cat/dog), we select a **threshold** value or tipping point above which we will classify values into class i and below which we classify values into class j.

There is a broad use of the Logistic regression algorithm in the field of machine learning. Logistic regression is also known in the literature as logit regression, maximum-entropy classification (MaxEnt) or the log-linear classifier.

[Sigmoid](https://ml-cheatsheet.readthedocs.io/en/latest/activation_functions.html#activation-sigmoid) function -

Notes:

* s(z) = output between 0 and 1 (probability estimate)
* z = the function input - the algorithm’s prediction which will be a linear equation. (y =mx + b . m is the weight vector for the independent variables vector x and b is the y-intercept)
* e = base of natural log

Graph:

A close up of a mans face

Description automatically generated

**An overview of the algorithm's parameters**

Logistic Regression uses the Log-Loss (AKA "Cross Entropy") in order to optimize its parameters, which is defined by the following function

When ' is the predicted probability for an instance, and is the true label. **N** - number of observations**.**

In order to avoid overfitting, we can add a penalty term called **regularization**. This regularization adds penalty as model complexity increases (in our case - when coefficients (w) gets too large). It is done by adding them to the total loss of the model**.**

**Parameters:**

**C**  - Inverse regularization parameter - A control variable that retains strength modification of Regularization by being inversely positioned to the Lambda regulator.

The relationship would be that lowering C - would strengthen the Lambda regulator.

**Regularization method -**As shown above - stands for abs distance, stands for quad\_distances

When using SKlearn (A machine learning library for the Python programming language) in order to make predictions with the Logistic Regression algorithm - There are several more parameters to pay attention too, I will mention a few:

**Tol -**  A stopping criteria

**Max\_iter** - Maximum number of iterations taken for the solvers to converge.

**Dual -** Dual or primal formulation. Dual formulation is only implemented for l2 penalty with liblinear solver. Prefer dual=False when n\_samples > n\_features.

**Fit\_intercept -** Specifies if a constant (a.k.a. bias or intercept) should be added to the decision function.

**Class\_weight -** Weights associated with classes in the form {class\_label: weight}. If not given, all classes are supposed to have weight one.

**How to use the algorithm**

In this process we will use machine learning libraries for the Python programming language.

1. First, in order to get a grip on the given data set behavior use exploration methods on the data set (visualization methods will be very beneficial). # An optional recommended step.

2. Define dependent / independent variables columns (Y\_test, X\_test) from the dataset.

3. Data Preprocessing - in order to gain reliable results using the Logistic Regression algorithm there is a need to perform several functions on the data set.

4. Use a method named 'Train\_test\_split' in order to divide your data into training data and validation data for both dependent and independent variables. This is an important stage which can help to prevent 'over fitting'. (now you will have 4 objects: Y\_val, X\_val, Y\_test, X\_test)

5. Use 'Grid search' -a function which perform hyper parameter tuning in order to determine the optimal values for a given model.

6. Define a Logistic Regression object using the parameters from step 5.

7. Activate the 'fit' method on the Logistic Regression object from step 6. **In a nutshell** - fittingis equal to training. Then, after it is trained, the model can be used to make predictions.

8. Use 'predict' method on the validation set (X\_val) - A technique performed on a database in order to produce predictions.

9. Use an evaluation method in order to measure the success (e.g. accuracy, misclass) of your prediction object from step 8 compare to the object Y\_val. For example, create a 'ROC curve' or a 'Confusion Matrix'.