**PYTHON – MACHINE LEARNING**

**Introduction**- Giving computers the ability to learn from Data

Machine learning evolved as a subfield of artificial intelligence that involved the development of self-learning algorithms to gain knowledge from that data to make predictions. Instead of requiring humans to manually derive rules and build models from analyzing large amounts of data, machine learning offers a more efficient alternative for capturing the knowledge in data to gradually improve the performance of predictive models and make data-driven decisions. Not only is machine learning becoming increasingly important in computer science research, but it also plays an ever-greater role in our everyday life.

**Three different types of machine Learning –**

Diagram

Description automatically generatedThe main goal in **supervised learning** is to learn a model from labeled training data that allows us to make predictions about unseen or future data. Here, the term supervised refers to a set of samples where the desired output signals (labels) are already known.

Diagram

Description automatically generatedThe example of **e-mail spam filtering**, we can train a model using a supervised machine learning algorithm on a corpus of labeled e-mail, e-mail that are correctly marked as spam or not-spam, to predict whether a new e-mail belongs to either of the two categories. A supervised learning task with discrete class labels, such as in the previous e-mail spam-filtering example, is also called a classification task.

* Another subcategory of supervised learning is regression, where the outcome signal is a continuous value:

**Classification for predicting class labels**

In machine learning, Classification, as the name suggests, classifies data into different parts/classes/groups. It is used to predict from which dataset the input data belongs to.

Classification is a subcategory of supervised learning where the goal is to predict the categorical class labels of new instances based on past observations. Those class labels are discrete, unordered values that can be understood as the group memberships of the instances. The previously mentioned example of e-mail-spam detection represents a typical example of a **binary classification task,** where the machine learning algorithm learns a set of rules to distinguish between two possible classes: spam and non-spam e-mail.

**Multi-class classification-** the set of class labels does not have to be of a binary nature. The predictive model learned by a supervised learning algorithm can assign any class label that was presented in the training dataset to a new, unlabeled instance. A typical **example** of a multi-class classification task is handwritten character recognition. Here, we could collect a training dataset that consists of multiple handwritten examples of each letter in the alphabet. Now, if a user provides a new handwritten character via an input device, our predictive model will be able to predict the correct letter in the alphabet with certain accuracy. However, our machine learning system would be unable to correctly recognize any of the digits zero to nine, for example, if they were not part of our training dataset.

Diagram

Description automatically generatedA white paper with writing on it

Description automatically generated with low confidence

Chart, scatter chart

Description automatically generatedGraphical user interface, text, email

Description automatically generated

The following figure illustrates the concept of a binary classification task given 30 training samples: 15 training samples are labeled as negative class (circles) and 15 training samples are labeled as positive class (plus signs). In this scenario, our dataset is two-dimensional, which means that each sample has two values associated with it: 1 x and 2 x. Now, we can use a supervised machine learning algorithm to learn a rule—the decision boundary represented as a black dashed line—that can separate those two classes and classify new data into each of those two categories given its 1 x and 2 x values: