

Unit 2

Machine Learning

Artificial Intelligence

School of Cyber Security & Digital Forensics

M. Sc. DFIS(Semester-II)

What is Machine Learning ?

- Machine Learning is the field of study that gives computers the ability to learn without being explicitly programmed. —Arthur Samuel, 1959
- 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

Types of machine learning systems

- **Whether or not they are trained with human supervision**

- **Supervised learning:** Tasks:- Classification and Prediction

(Popular algorithms: k-Nearest Neighbors, Linear Regression, Logistic Regression, Support Vector Machines, Decision Trees and Random Forests, Neural networks*)

- **Unsupervised learning:** Tasks:- Clustering , Anomaly detection, Novel detection & Association rule learning

(Popular algorithms: K-Means, DBSCAN, Hierarchical Cluster Analysis (HCA))

- **Semi-supervised learning:** Mostly combination of supervised and unsupervised learning

- **Reinforcement learning:** Learning system called **agent** learns best strategy based on **rewards** and **penalty**

Types of machine learning systems

- **Whether or not the system can learn incrementally from a stream of incoming data.**
 - Batch learning or offline learning
 - Online learning
- **Based on how they generalize.**
 - Instance-based learning
 - Model-based learning

Challenges of Machine Learning

- Data challenges
 - Insufficient Quantity of Training Data
 - Non-representative Training Data
 - Poor quality of data
 - Irrelevant features
- Algorithm challenges
 - Overfitting
 - Underfitting

Testing and Validating

- **Training set and Testing set**
- Generalization error or out-of-sample error : Error rate on new cases
- Testing set gives an estimate of generalization error
- If training error $<$ generalization error ; means overfitting
- **Hyperparameter Tuning and Model Selection**
- **Validation set** – if size too small cross validation

Machine learning project: Main steps

- Data Collection
- Discover and visualize data ...(Understanding datasets)
- Prepare data for machine learning algorithms ... (Data cleaning, Feature Encoding ,Feature Selection, Feature Normalization,)
- Select a model and train
- Fine tune model
- Present Model
- Launch, monitor and maintain the system

Note: *Reference for contents till this slide:* Chapter 1 of 2-Aurélien-Géron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow_-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-O'Reilly-Media-2019

Prepare data for machine learning algorithms

- Data Cleaning (source: 3.2 Han Kamber Data mining)

1. Missing Values

- i. Ignore the tuple
- ii. Fill missing values manually
- iii. Use a global constant
- iv. Use a measure of central tendency
- v. Use mean/median for all samples belonging to same class as given tuple
- vi. Use most probable value

Although we can try our best to clean the data after it is seized, good database and data entry procedure design should help minimize the number of missing values or errors in the first place.

Prepare data for machine learning algorithms

- Data Cleaning (source: 3.2 Han Kamber Data mining)

1. Missing Values
2. Noisy Data : **Noise** is a random error or variance in a measured variable

Following are data smoothing techniques:

Sorted data for *price* (in dollars): 4, 8, 15, 21, 21, 24, 25, 28, 34

- Binning** : (Equal width and Equal frequency) Smoothing by bin means,
Smoothing by bin medians, Smoothing by bin boundaries
- Regression**
- Outlier analysis**

Partition into (equal-frequency) bins:

Bin 1: 4, 8, 15

Bin 2: 21, 21, 24

Bin 3: 25, 28, 34

Smoothing by bin means:

Bin 1: 9, 9, 9

Bin 2: 22, 22, 22

Bin 3: 29, 29, 29

Smoothing by bin boundaries:

Bin 1: 4, 4, 15

Bin 2: 21, 21, 24

Bin 3: 25, 25, 34

Prepare data for machine learning algorithms

- **Feature Selection/ Attribute subset selection** (source: 3.4.4 Han Kamber Data mining) : reduces the data set size by **removing irrelevant** or **redundant** attributes (or dimensions). The goal of attribute subset selection is to find a minimum set of attributes such that the resulting probability distribution of the data classes is as close as possible to the original distribution obtained using all attributes.
- **Feature Selection:** The “best” (and “worst”) attributes are typically determined using tests of statistical significance, which assume that the attributes are independent of one another. Many other attribute evaluation measures can be used such as the information gain measure

Prepare data for machine learning algorithms

- **Feature Selection:** Following are some heuristic approaches:

1. Stepwise forward selection
2. Stepwise backward elimination
3. Combination of forward selection and backward elimination
4. Decision tree induction
5. Regression

Forward selection	Backward elimination	Decision tree induction
<p>Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$</p> <p>Initial reduced set: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ \Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$</p>	<p>Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$</p> <p>$\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ \Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$</p>	<p>Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$</p> <pre>graph TD; A4["A4?"] -- Y --> A1["A1?"]; A4 -- N --> A6["A6?"]; A1 -- Y --> C1_1((Class 1)); A1 -- N --> C2_1((Class 2)); A6 -- Y --> C1_2((Class 1)); A6 -- N --> C2_2((Class 2));</pre> <p>\Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$</p>

Prepare data for machine learning algorithms

- **Feature Normalization:** (Source :Section 3.5.2 Han Kamber) Normalizing the data attempts to give all attributes an equal weight. Normalization is particularly useful for classification algorithms involving neural networks or distance measurements such as nearest-neighbor classification and clustering. Following are some methods for normalization

1. Min-max normalization
$$v'_i = \frac{v_i - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A.$$

2. Z-score normalization
$$v'_i = \frac{v_i - \bar{A}}{\sigma_A},$$

3. Decimal Scaling
$$v'_i = \frac{v_i}{10^j},$$

where j is smallest integer such that $\max(|v'_i|) < 1$

Performance measures

- **Classification evaluation:**

- Accuracy = $(TP + TN) / (P + N)$
- Precision = $(TP) / (TP + FP)$
- Recall / Sensitivity / TPR = TP / P
- Specificity / TNR = TN / N
- F1-score = $(2 * \text{precision} * \text{recall}) / (\text{precision} + \text{recall})$
- Confusion Matrix is a tabular visualization of the **ground-truth labels versus model predictions**.

		Predicted	
		Positive (P)	Negative (N)
Actual	Positive (P)	TP	FN
	Negative (N)	FP	TN

Performance measures

- **Regression evaluation:**

- Mean Absolute Error = $\frac{\sum_{i=1}^N |\hat{y}_j - y_j|}{N}$

- Mean Squared Error = $\frac{\sum_{i=1}^N (\hat{y}_j - y_j)^2}{N}$

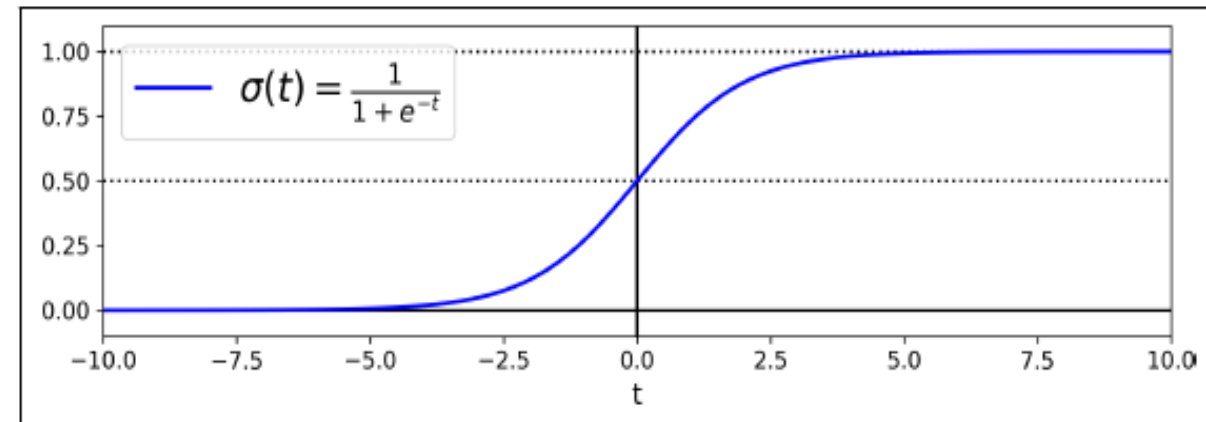
- Root Mean Squared Error = $\sqrt{\frac{\sum_{i=1}^N (\hat{y}_j - y_j)^2}{N}}$

Models of Supervised learning

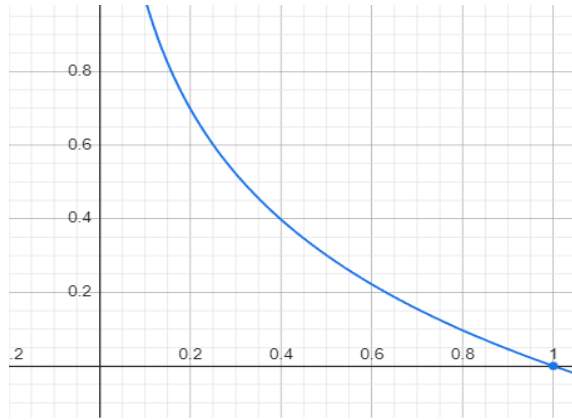
- Linear Regression: Prediction
- Logistic Regression: Classification
- k-Nearest Neighbour (kNN) : Classification
- Naïve Bayes : Classification
- Decision Trees : Classification
- Support Vector Machine (SVM) : Classification

Logistic Regression /Logit Regression

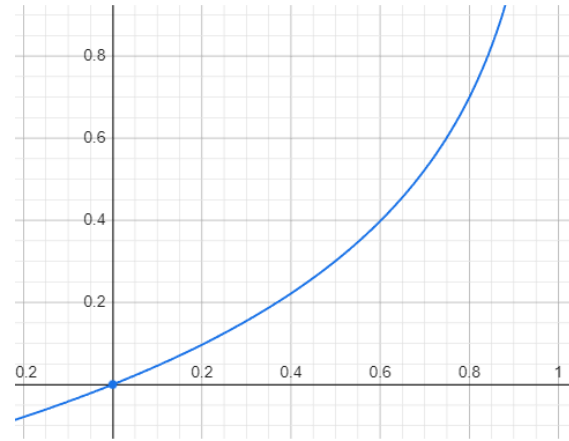
- Binary Classification problems: Email spam /not spam ; (Source: Hands on ML pg. no 144)
- Just like a Linear Regression model, a Logistic Regression model computes a weighted sum of the input features (plus a bias term), but instead of outputting the result directly like the Linear Regression model does, it outputs the *logistic* of this result $\hat{p} = \sigma(w^T x)$
- $\sigma(w^T x) = \frac{1}{1+e^{-w^T x}}$ Logistic / Sigmoid function (S-shaped)
- $\hat{y} = \begin{cases} 0, & \hat{p} < 0.5 \\ 1, & \hat{p} \geq 0.5 \end{cases}$



Logistic Regression /Logit Regression



Plot of x v/s $-\log(x)$



Plot of x v/s $-\log(1-x)$

- Cost for single instance =
$$\begin{cases} -\log(\hat{p}), & \text{if } y = 1 \\ -\log(1 - \hat{p}), & \text{if } y = 0 \end{cases}$$
- Cost over all training set (size N) =
$$\frac{1}{N} \sum -y \log(\hat{p}) - (1-y) \log(1 - \hat{p})$$

kNN Model

- No learning required; (Source: Machine learning mastery pg. no 98)
- KNN makes predictions using the training dataset directly. Predictions are made for a new data point by searching through the entire training set for the K most similar instances
- To determine which of the K instances in the training dataset are most similar to a new input a distance measure is used. For real-valued input variables, the most popular distance measure is **Euclidean distance**.

- $$dist(a, b) = \sqrt{\sum_{i=1}^N (a_i - b_i)^2}$$

kNN Model

- Instance –based learning ; Lazy learning
- If you are using K and you have an even number of classes (e.g. 2) it is a good idea to choose a K value with an odd number to avoid a tie. And the inverse, use an even number for K when you have an odd number of classes.

X1	X2	Y
7	7	B
7	4	B
3	4	G
1	4	G
3	7	?

Naïve Bayes Model

- Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called **class conditional independence**. (Source: Han Kamber Pg. 350 and Master Machine learning algorithm)
- Let D be a training set of tuples and their associated class labels. As usual, each tuple is represented by an n-dimensional attribute vector, X and represents each C_i class.
- $$P(C_i|X) = \frac{P(X|C_i) P(C_i)}{P(X)} \text{ maximize } P(C_i|X)$$
- *Laplacian correction*

Naïve Bayes Model

- *Weather = sunny, Car = Working, class = ?*

Weather	Car	Class
sunny	working	go-out
rainy	broken	go-out
sunny	working	go-out
sunny	working	go-out
sunny	working	go-out
rainy	broken	stay-home
rainy	broken	stay-home
sunny	working	stay-home
sunny	broken	stay-home
rainy	broken	stay-home