PS03EMCA58 Machine Learning (Units 2 & 3)

Dr. J. V. Smart

Table of Contents

Syllabus

Syllabus with effect from the Academic Year 2021-2022

Course Code	PS03EMCA58	Title of the Course	MACHINE LEARNING
Total Credits of the Course	4	Hours per Week	4

Course Objectives:	1. To learn the fundamental issues and challenges of traditional artificial intelligence systems and the need for machine learning. 2. To learn the strengths and weaknesses of many popular machine learning approaches. 3. TO learn various machine learning algorithms, paradigms of supervised and unsupervised learning, and hybrid computational intelligence techniques.

	4. To be able to learn applications of machine learning in various real-life systems.

Course Content

Unit	Description	Weightage* (%)
1.	**Introduction to Machine Learning** - Types of learning: Human and machine learning - Types of machine learning - Applications of machine learning - Tools for machine	25
2.	**Supervised Learning** - Introduction and examples of supervised learning - Classification model and classification learning steps - Training data sets and validation data sets - Deep learning in artificial neural network, Deep Vs. shallow learning - Introduction to deep learning	25
3.	**Unsupervised Learning** - Introduction to Clustering - Self Organizing map/Kohenon neural network - K nearest neighborhood - K-means and its variations - Applications of unsupervised learning - Introduction to hybrid	25
4.	**Hybrid Computational Intelligence** - Constituents of computational intelligence - Possible hybridization of constituents of computational intelligence - Neuro-Fuzzy Systems, Neuro-Genetic Systems and Neuro-Fuzzy- Genetic systems - Applications of computational intelligence system in real	25

Teaching-Learning Methodology	Blended learning approach incorporating traditional classroom teaching as well as online / ICT-based teaching practices

Evaluation Pattern

Sr. No.	Details of the Evaluation	Weightage
1.	Internal Written / Practical Examination (As per CBCS R.6.8.3)	15%
2.	Internal Continuous Assessment in the form of Practical, Viva-voce, Quizzes, Seminars, Assignments, Attendance (As per CBCS R.6.8.3)	15%

Sr. No.	Details of the Evaluation	Weightage
3.	University Examination	70%

Course Outcomes: Having completed this course, the learner will be able to	
1.	To understand the fundamental concepts related to machine learning techniques.
2.	To understanding the application of modern intelligent systems in solving real-life problems.

Suggested References:

Sr. No.	References
1.	Saikat Dutt, Subramanian Chandramouli, Amit Kumar Das, “Machine Learning”, Pearson Education, 2018.
2.	Shai Shalev-Shwartz and Shai Ben-David, Understanding Machine Learning: From Theory to Algorithms, Cambridge University Press, 2014.
3.	Sajja P S, Illustarted computational Intellignce:Examples and Applications, Springer International Publishing, 2020.
4.	Christopher M. Bishop, Pattern recognition and machine learning, Spinger, 2006.
5.	Ethem Alpaydın, Introduction to Machine Learning, Second Edition, 2010.

Machine Learning

Generally, a computer does what it is programmed to do — nothing more and nothing less. However, Machine learning programs can perform tasks without being explicitly programmed to do so. Machine learning is a branch of artificial intelligence. It involves computers learning from data provided so that they carry out certain tasks. Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so. There are various approaches to machine learning.

Supervised Learning

Supervised learning is a type of machine learning in which the machine learning algorithm is trained using a set of *labelled* training data called the training data set. The training data set is input data tagged with the expected output. The training generates a model. The model is then used by the supervised learning algorithm to predict / produce output from input. The model's functioning and accuracy can be tested using one or more test data sets / validation data sets.

Linear Regression

Example-1 Using the Scikit-learn Standard Dataset Diabetes

Scikit-learn is a popular machine learning library for Python. It is based on NumPy and SciPy.

The Diabetes dataset is one of the preinstalled *toy datasets* in Scikit-learn. It is a standard dataset for learning linear regression using Scikit-learn. The original dataset is available at Diabetes Data. The dataset is based on a study of 442 patients of diabetes. It contains data of ten baseline variables — age, sex, body mass index, average blood pressure, and six blood serum measurements for each patient, It also contains a quantitative measure of disease progression (increase in disease) one year after baseline. The dataset can be used to measure the correlation between one or more of the 10 parameters and disease progression.

Sample Data from the Diabetes Dataset

AGE	SEX	BMI	BP	S1	S2	S3	S4	S5	S6	Disease Progression
59	2	32.1	101	157	93.2	38	4	4.8598	87	151
48	1	21.6	87	183	103.2	70	3	3.8918	69	75
72	2	30.5	93	156	93.6	41	4	4.6728	85	141
24	1	25.3	84	198	131.4	40	5	4.8903	89	206
50	1	23	101	192	125.4	52	4	4.2905	80	135

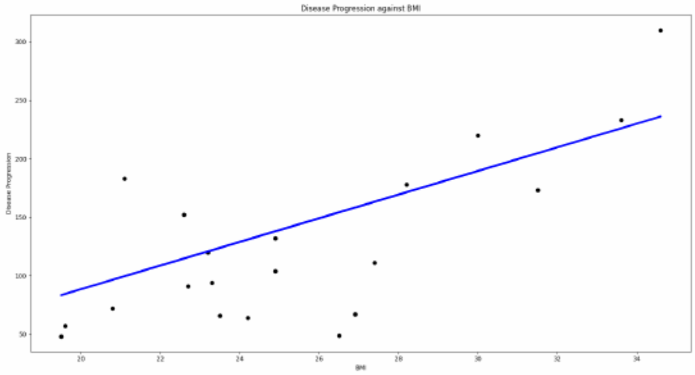
In this example, we shall study the correlation between BMI and disease progression.

BMI

Body Mass Index. It is calculated as body mass (weight) in kg divided by the square of the body height in meters

Disease Progression

In this study, this is a number representing increase in disease in one year. Lower numbers indicate smaller increase in disease, while larger number indicate higher increase in disease



Diabetes Disease Progression against BMI

```
#Code source: Jaques Grobler
#License: BSD 3 clause

import matplotlib.pyplot as plt
import numpy as np
from sklearn import datasets, linear_model
from sklearn.metrics import mean_squared_error, r2_score

#Load the diabetes dataset
diabetes_X, diabetes_y = datasets.load_diabetes(return_X_y=True)

diabetes_X = diabetes_X[:, np.newaxis, 2]
```

```
bmi = [ [ 32.1 ], [ 21.6 ], [ 30.5 ], [ 25.3 ], [ 23 ], [ 22.6 ], [ 22 ],
        [ 26.2 ], [ 32.1 ], [ 30 ], [ 18.6 ], [ 28 ], [ 23.7 ], [ 26.2 ],
        [ 24 ], [ 24.7 ], [ 30.3 ], [ 27.5 ], [ 25.4 ], [ 24.7 ], [ 21.1
        ], [ 24.3 ], [ 26 ], [ 32 ], [ 29.7 ], [ 25.2 ], [ 19.2 ], [ 31.9
        ], [ 24.4 ], [ 25.8 ], [ 30.5 ], [ 20.3 ], [ 38 ], [ 21.7 ], [
        20.5 ], [ 23.5 ], [ 28.5 ], [ 27.4 ], [ 33 ], [ 27.7 ], [ 25.6 ],
        [ 20.1 ], [ 25.4 ], [ 24.2 ], [ 32.7 ], [ 23.1 ], [ 25.3 ], [ 19.6
        ], [ 22.5 ], [ 27.7 ], [ 25.7 ], [ 27.9 ], [ 25.5 ], [ 24.9 ], [
        28.7 ], [ 21.8 ], [ 30.2 ], [ 20.5 ], [ 20.4 ], [ 24 ], [ 26 ], [
        26.8 ], [ 25.7 ], [ 22.9 ], [ 24 ], [ 24.1 ], [ 24.7 ], [ 25 ], [
        23.6 ], [ 22.1 ], [ 19.9 ], [ 29.5 ], [ 26 ], [ 24.5 ], [ 26.6 ],
        [ 23.5 ], [ 29 ], [ 23 ], [ 21 ], [ 22.9 ], [ 27.5 ], [ 24.3 ], [
        23.1 ], [ 27.3 ], [ 22.7 ], [ 33 ], [ 19.4 ], [ 25.8 ], [ 22.6 ],
        [ 21.9 ], [ 24 ], [ 31.2 ], [ 26.8 ], [ 20.4 ], [ 24.8 ], [ 21 ],
        [ 27.3 ], [ 34.6 ], [ 25.9 ], [ 20.4 ], [ 28 ], [ 22.2 ], [ 29 ],
        [ 30.2 ], [ 32.4 ], [ 23.4 ], [ 19.3 ], [ 31 ], [ 30.6 ], [ 25.5
        ], [ 23.4 ], [ 26.8 ], [ 28.3 ], [ 27.7 ], [ 36.6 ], [ 26.5 ], [
        31.8 ], [ 24.4 ], [ 25.4 ], [ 22 ], [ 26.8 ], [ 28 ], [ 33.9 ], [
        29.6 ], [ 28.6 ], [ 25.6 ], [ 20.7 ], [ 26.2 ], [ 20.6 ], [ 27.9
        ], [ 35.3 ], [ 19.9 ], [ 24.4 ], [ 21.4 ], [ 30.4 ], [ 31.6 ], [
        18.8 ], [ 31 ], [ 36.7 ], [ 32.1 ], [ 27.7 ], [ 30.8 ], [ 27.5 ],
        [ 26.9 ], [ 30.7 ], [ 38.3 ], [ 31.9 ], [ 35 ], [ 27.8 ], [ 25.9
        ], [ 32.9 ], [ 26 ], [ 26.3 ], [ 22.3 ], [ 28.3 ], [ 32 ], [ 25.4
        ], [ 23.3 ], [ 20.3 ], [ 30.4 ], [ 20.6 ], [ 32.3 ], [ 29.2 ], [
        33.1 ], [ 24.6 ], [ 20.2 ], [ 20.8 ], [ 32.8 ], [ 31.9 ], [ 23.9
        ], [ 24.5 ], [ 22.1 ], [ 33 ], [ 19 ], [ 27.3 ], [ 22.8 ], [ 28.2
        ], [ 28.9 ], [ 25.6 ], [ 24.9 ], [ 26.8 ], [ 22.4 ], [ 26.9 ], [
        23.1 ], [ 28.6 ], [ 24.7 ], [ 30.3 ], [ 21.3 ], [ 26.1 ], [ 20.2
        ], [ 25.2 ], [ 22.5 ], [ 23.5 ], [ 25.9 ], [ 20.9 ], [ 28.7 ], [
        22.1 ], [ 26.7 ], [ 31.4 ], [ 22.2 ], [ 21 ], [ 21.2 ], [ 26.5 ],
        [ 29.2 ], [ 27 ], [ 30.7 ], [ 28.8 ], [ 30.6 ], [ 30.1 ], [ 24.7
        ], [ 27.7 ], [ 29.8 ], [ 26.7 ], [ 19.8 ], [ 23.3 ], [ 35.1 ], [
        29.7 ], [ 29.3 ], [ 20.3 ], [ 22.5 ], [ 22.7 ], [ 22.8 ], [ 24 ],
        [ 24.2 ], [ 20.2 ], [ 29.4 ], [ 22.1 ], [ 23.6 ], [ 25.2 ], [ 24.9
        ], [ 33 ], [ 23.5 ], [ 26.4 ], [ 29.8 ], [ 30 ], [ 25 ], [ 27 ], [
        20 ], [ 25.5 ], [ 28.2 ], [ 33.3 ], [ 25.6 ], [ 24.2 ], [ 22.1 ],
        [ 31.4 ], [ 23.1 ], [ 23.4 ], [ 18.8 ], [ 30.8 ], [ 32 ], [ 31.6
        ], [ 35.5 ], [ 31.9 ], [ 29.5 ], [ 31.6 ], [ 20.3 ], [ 41.3 ], [
        21.2 ], [ 24.1 ], [ 23 ], [ 25.6 ], [ 22.5 ], [ 38.2 ], [ 19.2 ],
        [ 29 ], [ 24 ], [ 20.6 ], [ 26.3 ], [ 34.6 ], [ 23.4 ], [ 29.2 ],
        [ 27.2 ], [ 27 ], [ 24.5 ], [ 24.1 ], [ 25.3 ], [ 28.8 ], [ 20.9
        ], [ 23 ], [ 24.1 ], [ 28.1 ], [ 18 ], [ 25.9 ], [ 21.5 ], [ 24.3
        ], [ 24.5 ], [ 21.3 ], [ 25.8 ], [ 24.8 ], [ 31.5 ], [ 33.5 ], [
        28.1 ], [ 24.3 ], [ 35 ], [ 23.5 ], [ 30 ], [ 20.7 ], [ 25.6 ], [
        22.9 ], [ 25.1 ], [ 33.2 ], [ 24.1 ], [ 29.5 ], [ 29.6 ], [ 22.8
        ], [ 22.7 ], [ 26.2 ], [ 23.5 ], [ 22.1 ], [ 26.5 ], [ 32.4 ], [
        30.1 ], [ 24.2 ], [ 31.3 ], [ 30.1 ], [ 24.5 ], [ 27.7 ], [ 23.2
        ], [ 27 ], [ 26.8 ], [ 29.2 ], [ 31.2 ], [ 32.1 ], [ 25.7 ], [
        26.9 ], [ 31.4 ], [ 25.6 ], [ 37 ], [ 32.6 ], [ 21.2 ], [ 29.2 ],
        [ 24 ], [ 36.1 ], [ 25.8 ], [ 22 ], [ 21.9 ], [ 34.3 ], [ 25.2 ],
        [ 23.3 ], [ 25.7 ], [ 25.1 ], [ 31.9 ], [ 28.4 ], [ 28.1 ], [ 25.3
        ], [ 26.1 ], [ 28 ], [ 23.6 ], [ 24.5 ], [ 21 ], [ 32 ], [ 22.6 ],
        [ 19.7 ], [ 21.2 ], [ 30.6 ], [ 25.5 ], [ 23.3 ], [ 31 ], [ 18.5
        ], [ 26.9 ], [ 28.3 ], [ 25.7 ], [ 36.1 ], [ 24.1 ], [ 25.8 ], [
        22.8 ], [ 39.1 ], [ 42.2 ], [ 26.6 ], [ 29.9 ], [ 21 ], [ 25.5 ],
        [ 24.2 ], [ 25.4 ], [ 23.2 ], [ 26.1 ], [ 32.7 ], [ 27.3 ], [ 26.6
        ], [ 22.8 ], [ 28.8 ], [ 18.1 ], [ 32 ], [ 23.7 ], [ 23.6 ], [
        24.6 ], [ 22.6 ], [ 27.8 ], [ 24.1 ], [ 26.5 ], [ 32.8 ], [ 19.9
        ], [ 23.6 ], [ 22.1 ], [ 28.1 ], [ 26.5 ], [ 23.5 ], [ 26 ], [
        27.8 ], [ 28.5 ], [ 30.6 ], [ 22.2 ], [ 23.3 ], [ 35.4 ], [ 31.4
        ], [ 37.8 ], [ 18.9 ], [ 35 ], [ 21.7 ], [ 25.3 ], [ 23.8 ], [
        31.8 ], [ 34.3 ], [ 26.3 ], [ 27 ], [ 27.2 ], [ 33.8 ], [ 33 ], [
        24.1 ], [ 21.3 ], [ 23 ], [ 27.9 ], [ 33.6 ], [ 22.7 ], [ 27.4 ],
        [ 22.6 ], [ 23.2 ], [ 26.9 ], [ 34.6 ], [ 23.3 ], [ 21.1 ], [ 23.5
        ], [ 31.5 ], [ 20.8 ], [ 26.5 ], [ 24.2 ], [ 19.5 ], [ 28.2 ], [
        24.9 ], [ 24.9 ], [ 30 ], [ 19.6 ] ]
diabetes_X = bmi
```

```
diabetes_y = [ 151, 75, 141, 206, 135, 97, 138, 63, 110, 310, 101, 69,
        179, 185, 118, 171, 166, 144, 97, 168, 68, 49, 68, 245, 184, 202,
        137, 85, 131, 283, 129, 59, 341, 87, 65, 102, 265, 276, 252, 90,
        100, 55, 61, 92, 259, 53, 190, 142, 75, 142, 155, 225, 59, 104,
        182, 128, 52, 37, 170, 170, 61, 144, 52, 128, 71, 163, 150, 97,
        160, 178, 48, 270, 202, 111, 85, 42, 170, 200, 252, 113, 143, 51,
        52, 210, 65, 141, 55, 134, 42, 111, 98, 164, 48, 96, 90, 162, 150,
        279, 92, 83, 128, 102, 302, 198, 95, 53, 134, 144, 232, 81, 104,
        59, 246, 297, 258, 229, 275, 281, 179, 200, 200, 173, 180, 84,
        121, 161, 99, 109, 115, 268, 274, 158, 107, 83, 103, 272, 85, 280,
        336, 281, 118, 317, 235, 60, 174, 259, 178, 128, 96, 126, 288, 88,
        292, 71, 197, 186, 25, 84, 96, 195, 53, 217, 172, 131, 214, 59,
        70, 220, 268, 152, 47, 74, 295, 101, 151, 127, 237, 225, 81, 151,
        107, 64, 138, 185, 265, 101, 137, 143, 141, 79, 292, 178, 91, 116,
        86, 122, 72, 129, 142, 90, 158, 39, 196, 222, 277, 99, 196, 202,
        155, 77, 191, 70, 73, 49, 65, 263, 248, 296, 214, 185, 78, 93,
        252, 150, 77, 208, 77, 108, 160, 53, 220, 154, 259, 90, 246, 124,
        67, 72, 257, 262, 275, 177, 71, 47, 187, 125, 78, 51, 258, 215,
        303, 243, 91, 150, 310, 153, 346, 63, 89, 50, 39, 103, 308, 116,
        145, 74, 45, 115, 264, 87, 202, 127, 182, 241, 66, 94, 283, 64,
        102, 200, 265, 94, 230, 181, 156, 233, 60, 219, 80, 68, 332, 248,
        84, 200, 55, 85, 89, 31, 129, 83, 275, 65, 198, 236, 253, 124, 44,
        172, 114, 142, 109, 180, 144, 163, 147, 97, 220, 190, 109, 191,
        122, 230, 242, 248, 249, 192, 131, 237, 78, 135, 244, 199, 270,
        164, 72, 96, 306, 91, 214, 95, 216, 263, 178, 113, 200, 139, 139,
        88, 148, 88, 243, 71, 77, 109, 272, 60, 54, 221, 90, 311, 281,
        182, 321, 58, 262, 206, 233, 242, 123, 167, 63, 197, 71, 168, 140,
        217, 121, 235, 245, 40, 52, 104, 132, 88, 69, 219, 72, 201, 110,
        51, 277, 63, 118, 69, 273, 258, 43, 198, 242, 232, 175, 93, 168,
        275, 293, 281, 72, 140, 189, 181, 209, 136, 261, 113, 131, 174,
        257, 55, 84, 42, 146, 212, 233, 91, 111, 152, 120, 67, 310, 94,
        183, 66, 173, 72, 49, 64, 48, 178, 104, 132, 220, 57 ]

print()
print('Total data points in the dataset: {0}'.format(len(diabetes_X)))
print()

#Split the data into training/testing sets
#Train the model using all but the last 20 values
#Test / validate the model using last 20 values
diabetes_X_train = diabetes_X[:-20]
print('Training set size:', len(diabetes_X_train))
diabetes_X_test = diabetes_X[-20:]
print('Test set size:', len(diabetes_X_test))
print()

#Split the targets into training/testing sets
diabetes_y_train = diabetes_y[:-20]
diabetes_y_test = diabetes_y[-20:]

#Create linear regression object
regr = linear_model.LinearRegression()

#Train the model using the training sets
regr.fit(diabetes_X_train, diabetes_y_train)

#Make predictions using the testing set
diabetes_y_pred = regr.predict(diabetes_X_test)

print("Coefficient for BMI: {0:5.2f}".format(regr.coef_[0]))
print("Interpretation: For an increase of 1 in BMI, there is an increase
        of {0:5.2f} in disease progression".format(regr.coef_[0]))
print()

#The mean squared error
```

```
#print("Mean squared error: %.2f" % mean_squared_error(diabetes_y_test,
        diabetes_y_pred))

#The coefficient of determination: 1 is perfect prediction
print("Coefficient of determination (r2_score): %.2f" %
        r2_score(diabetes_y_test, diabetes_y_pred))
print("Interpretation: a value of 1 indicates perfect prediction. A small
        or negative value indicates inaccurate prediction")
print()

#Plot outputs
plt.scatter(diabetes_X_test, diabetes_y_test, color="black")
plt.title("Disease Progression against BMI")
plt.plot(diabetes_X_test, diabetes_y_pred, color="blue", linewidth=3)
plt.xlabel('BMI')
plt.ylabel('Disease Progression')

#plt.xticks(())
#plt.yticks(())

plt.show()
/////// OUTPUT ///////
Total data points in the dataset: 442

Training set size: 422
Test set size: 20

Coefficient for BMI: 10.11
Interpretation: For an increase of 1 in BMI, there is an increase of 10.11
        in disease progression

Coefficient of determination (r2_score): 0.47
Interpretation: a value of 1 indicates perfect prediction. A small or
        negative value indicates inaccurate prediction
```

Decision Trees

Attribute Selection Measures (ASMs)

Entropy and Information Gain

Entropy and Information Content

Entropy is the amount of unorderliness, uncertainty or surprise in a system. It is also a measure of information in the system, because deviation from the "normal" or "standard" values can be used to convey information. The symbol `H` (Capital Eta) is used for entropy. In information science, the unit of entropy is usually `bit`.

Serial Communication:

High Voltage : 1 Low Voltage : 0

```
 __________________________________
|

1111111111...
```

```
   __    ____    __      __     ______      __
__|  |__|    |__|  |____|  |__|       |____|  |

01011010010111001
```

Calculating Entropy

$$H(X) = -\sum_{i=1}^{n} P(x_i) log P(x_i)$$

When the base of the logarithm is 2, the unit of entropy / information is bit.

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_2 P(x_i) \, bits$$

Toss of 1 unbiased coin

$$P(T) \rightarrow \frac{1}{2}$$

$$P(H) \rightarrow \frac{1}{2}$$

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_2 P(x_i)$$

$$H = -(P(H) log_2 P(H) + P(T) log_2 P(T))$$

$$H = -(\frac{1}{2} log_2 \frac{1}{2} + \frac{1}{2} log_2 \frac{1}{2})$$

$$H = -(\frac{1}{2} log_2 2^{-1} + \frac{1}{2} log_2 2^{-1})$$

$$H = -(\frac{1}{2}(-1) + \frac{1}{2}(-1))$$

$$H = -(-\frac{1}{2} + -\frac{1}{2})$$

$$H = -(-1)$$

$$H = 1 \, bit$$

The result of the toss of a single unbiased coin provides 1 bit of information.

```
0=T
1=H
```

Toss of 2 unbiased coins

$$P(TT) \rightarrow \frac{1}{4}$$

$$P(TH) \rightarrow \frac{1}{4}$$

$$P(HT) \rightarrow \frac{1}{4}$$

$$P(HH) \rightarrow \frac{1}{4}$$

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_2 P(x_i)$$

$$H = -(P(TT)log_2P(TT) + P(TH)log_2P(TH) + P(HT)log_2P(HT) + P(HH)log_2P(HH))$$

$$H = -(\frac{1}{4}log_2\frac{1}{4} + \frac{1}{4}log_2\frac{1}{4} + \frac{1}{4}log_2\frac{1}{4} + \frac{1}{4}log_2\frac{1}{4})$$

$$H = -(\frac{1}{4}log_2 2^{-2} + \frac{1}{4}log_2 2^{-2} + \frac{1}{4}log_2 2^{-2} + \frac{1}{4}log_2 2^{-2})$$

$$H = -(\frac{1}{4}(-2) + \frac{1}{4}(-2) + \frac{1}{4}(-2) + \frac{1}{4}(-2))$$

$$H = -(-\frac{1}{2} + -\frac{1}{2} + -\frac{1}{2} + -\frac{1}{2})$$

$$H = -(-2)$$

$$H = 2\,bits$$

The result of the toss of two unbiased coins provides 2 bits of information.

```
00=TT
01=TH
10=HT
11=HH
```

Toss of 1 fully biased coin and 1 unbiased coin

$$P(TT) \rightarrow 0$$

$$P(TH) \rightarrow 0$$

$$P(HT) \rightarrow \frac{2}{4}$$

$$P(HH) \rightarrow \frac{2}{4}$$

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_2 P(x_i)$$

$$H = -(P(TT)log_2P(TT) + P(TH)log_2P(TH) + P(HT)log_2P(HT) + P(HH)log_2P(HH))$$

$$H = -(0 + 0 + \frac{1}{2}log_2\frac{1}{2} + \frac{1}{2}log_2\frac{1}{2})$$

$$H = -(0 + 0 + \frac{1}{2}log_2 2^{-1} + \frac{1}{2}log_2 2^{-1})$$

$$H = -(0 + 0 + \frac{1}{2}(-1) + \frac{1}{2}(-1))$$

$$H = -(0 + 0 + -\frac{1}{2} + -\frac{1}{2})$$

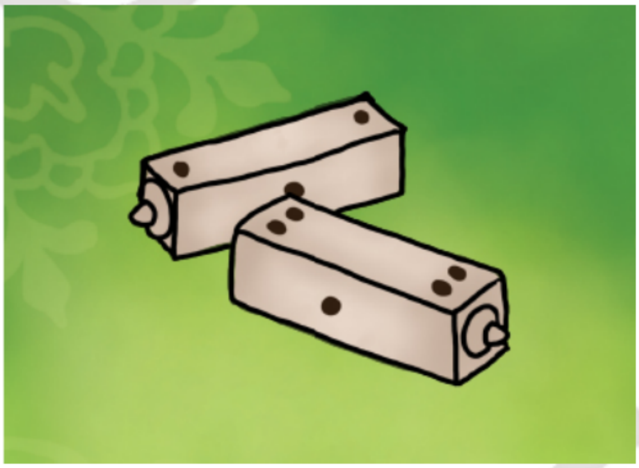
$$H = -(-1)$$

$$H = 1\, bit$$

The result of the toss of one fully biased and one unbiased coin provides 1 bit of information.

```
0=HT
1=HH
```

Rolling of an Unbiased Stick Die



Stick Dice

$$P(1) \rightarrow \frac{1}{4}$$

$$P(2) \rightarrow \frac{1}{4}$$

$$P(3) \rightarrow \frac{1}{4}$$

$$P(4) \rightarrow \frac{1}{4}$$

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_2 P(x_i)$$

$$H = -(P(1)log_2P(1) + P(2)log_2P(2) + P(3)log_2P(3) + P(4)log_2P(4))$$

$$H = -(\frac{1}{4}log_2\frac{1}{4} + \frac{1}{4}log_2\frac{1}{4} + \frac{1}{4}log_2\frac{1}{4} + \frac{1}{4}log_2\frac{1}{4})$$

$$H = -(\frac{1}{4}log_2 2^{-2} + \frac{1}{4}log_2 2^{-2} + \frac{1}{4}log_2 2^{-2} + \frac{1}{4}log_2 2^{-2})$$

$$H = -(\frac{1}{4}(-2) + \frac{1}{4}(-2) + \frac{1}{4}(-2) + \frac{1}{4}(-2))$$

$$H = -(-\frac{1}{2} + -\frac{1}{2} + -\frac{1}{2} + -\frac{1}{2})$$

$$H = -(-2)$$

$$H = 2\,bits$$

The result of rolling an unbiased stick die provides 2 bits of information.

```
00=1
01=2
10=3
11=4
```

Rolling of a Biased Stick Die

$$P(1) \rightarrow \frac{1}{6}$$

$$P(2) \rightarrow \frac{1}{6}$$

$$P(3) \rightarrow \frac{3}{6}$$

$$P(4) \rightarrow \frac{1}{6}$$

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_2 P(x_i)$$

$$H = -(P(1)log_2 P(1) + P(2)log_2 P(2) + P(3)log_2 P(3) + P(4)log_2 P(4))$$

$$H = -(\frac{1}{6}log_2 \frac{1}{6} + \frac{1}{6}log_2 \frac{1}{6} + \frac{3}{6}log_2 \frac{3}{6} + \frac{1}{6}log_2 \frac{1}{6})$$

$$H = -((0.1667 \times -2.585) + (0.1667 \times -2.585) + (0.5 \times (-1)) + (0.1667 \times -2.585))$$

$$H = -((-0.4309) + (-0.4309) + (-0.5) + (-0.4309))$$

$$H = -(-1.793)$$

$$H = 1.793\, bits$$

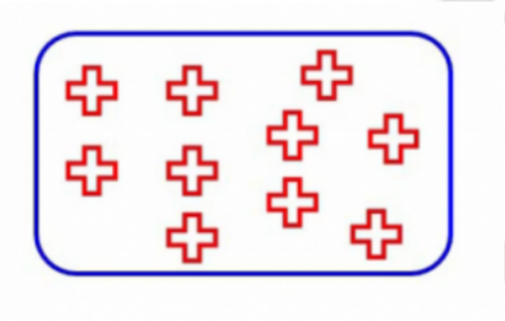
The result of rolling the biased stick die as discussed above provides 1.793 bits of information.

Gini Index and Gini Impurity

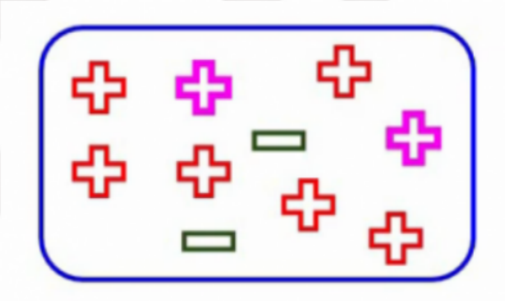
Gini Impurity

Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset. Its value ranges between 0 and 1.

Gini impurity measures the impurity of a node. A node having elements belonging to a single class is considered pure, while a node having elements belonging to multiple classes is considered impure.



A Pure Node



An Impure Node

If all the elements in a node belong to a single class, then the Gini impurity for such a node is 0.

If all the elements in a node are randomly distributed across the classes, then the Gini impurity for such a node is 1.

If all the elements in a node are equally distributed across the classes, then the Gini impurity for such a node is 0.5.

$$I_G(p) = \sum_{i=1}^{n} p_i(1 - p_i)$$

$$I_G(p) = 1 - \sum_{i=1}^{n} p_i^2$$

Compared to entropy calculations and the information gain measure, Gini impurity is computationally less intensive, i.e. it requires less computations.

Random Forest

- Ensemble learning - multiple algorithms or same algorithm trained on multiple datasets
- Multiple random selections of a subset of training dataset are made and decision trees are created for each subset
- Random sampling with replacement
- Forest - multiple trees
- When predicting, mean / average of the predictions of all the decision trees is used or majority wins rule is used
- Random forest is used to solve the problem of overfitting that occurs with single decision tree
- The process of selecting multiple random subsets from the training dataset is known as bootstrap aggregating or bagging

k-nearest neighbors

Support Vector Machine (SVM)

Unsupervised Learning

Notes

dependent variables v/s independent variables