

# Homework 2 - Entropy, Cross-Entropy, KL Divergence

In [3]:

```
import numpy as np
```

## Information Theory

Bit information is used with 1s and 0s. This bit information is used for reduction of the uncertainty of the information with the factor of 2. It sounds complicated. So how elaborate this?

Think of a person you have never met. What information you need to know. Maybe you want to their gender or they are old or young. Assuming this information comes binary and at the equal probability. If we want to encode this information with 0 and 1's,



shutterstock.com · 1190374711

the bit size is the total number of random variables factored by two:

- Bit size:  $\log(2) = 1$  bit

Meaning that for male, we encode **male** as '0' while encode **women** as '1'. But if we ask what is the informativeness of gender to you? Then we need **entropy**

## Entropy

Entropy is the average information you receive from a distribution of given occurrences. Entropy is calculated by a negative sum of the probability of event multiplied by the  $\log_2$  of probability, meaning that with the true possibility of all events' occurrence how much information can be received. Information is larger when the probability is smaller because  $\log_2$

$$H(p) = - \sum_i p_i * \log_2(p_i)$$

In [4]:

```
probability_of_gender =[0.5, 0.5]
```

In [5]:

```
def calculate_entropy(prob):  
    return -np.array([i*np.log2(i) for i in prob]).sum()
```

In [6]:

```
calculate_entropy(probability_of_gender)
```

Out[6]:

1.0

But if we say that the distributions are not equal and it is more likely for you encounter a female than a male?

In [7]:

```
probability_of_gender =[0.75, 0.25]
```

In [8]:

```
calculate_entropy(probability_of_gender)
```

Out[8]:

0.8112781244591328

Then we see that bit-wise information reduces since the your chance of encountering a female is more likely, your uncertainty of the person's gender gets smaller.

But what do you think the gender of that person? If you want to find your expectation meets with the reality, you need **cross-entropy**

## Cross Entropy

Cross entropy on the other hand is **the average message length** from a distribution of given probabilities. Cross entropy is calculated by a negative sum of actual probability of events multiplied by the predicted informativeness  $\log_2(p(event))$  of events. This gives the expected information length occurred via your prediction.

$$H(p, q) = - \sum_i p_i * \log_2(q_i)$$

In [9]:

```
def calculate_cross_entropy(true_prob, predict_prob):  
    return -np.array([t*np.log2(p) for t,p in zip(true_prob, predict_prob)]).sum()
```

In [10]:

```
probability_of_gender = [0.5, 0.5]
prediction_of_gender = [0.75, 0.25]
```

In [11]:

```
calculate_cross_entropy(probability_of_gender, prediction_of_gender)
```

Out[11]:

1.207518749639422

There we can say that our expectations haven't met with the true probability of gender. Your uncertainty of The difference of the informativeness of the gender and the informativeness of the expectation of gender gives us the KL Divergence.

## KL Divergence

KL divergence can be calculated as the negative sum of probability of each event in P multiplied by the log of the probability of the event in Q over the probability of the event in P.

$$D_{\text{KL}}(p \parallel q) = H(p, q) - H(p)$$

In [12]:

```
def calculate_kl_divergance(true_prob, predict_prob):
    return calculate_cross_entropy(true_prob, predict_prob) - calculate_entropy(tru
```

In [13]:

```
calculate_kl_divergance(probability_of_gender, prediction_of_gender)
```

Out[13]:

0.20751874963942196