Information Theory

Araştırmacılar, 1900'lerin başından beri bilgiyi nicelleştirme üzerine kafa yordular ve 1948'de Claude Shannon, "A Mathematical Theory of Communication" adlı olağanüstü bir makale yayınladı. Bu makale Bilgi Teorisi alanını doğurdu. Bilgi Teorisi, tanım olarak, bilginin nicelenmesi, depolanması ve iletişiminin incelenmesidir. Ama bundan çok daha fazlası. İstatistiksel Fizik, Bilgisayar Bilimleri, Ekonomi vb. Alanlara önemli katkılar sağlamıştır.

Shannon'ın makalesinin ana odak noktası, makaleyi yayınladığında Bell Laboratuvarlarında çalıştığı için genel iletişim sistemiydi. Bilgi entropisi ve fazlalık gibi birkaç önemli kavram oluşturdu. Günümüzde temel temelleri kayıpsız veri sıkıştırma, kayıplı veri sıkıştırma ve kanal kodlama alanlarında uygulanmaktadır.

Bilgi Teorisinde kullanılan teknikler doğası gereği olasılıklıdır ve genellikle 2 spesifik nicelik ile ilgilenir, yani. Entropi ve Karşılıklı Bilgi. Bu iki terime daha derin bir dalış yapalım.

Shannon Entropy (or just Entropy)

Entropy is the measure of uncertainty of a random variable or the amount of information required to describe a variable. Suppose x is a discrete random variable, and it can take any value defined in the set, χ. Let's assume the set is finite in this scenario. The probability distribution for x will be $p(x) = \Pr\{\chi = x\}, x \in \chi$. With this in mind, entropy can be defined as

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log p(x).$$

Entropy

The unit of entropy is bit. If you observe the formula, entropy is entirely dependent on the **probability of the random variable** and not on the value of x itself. There is a negative sign in front of the formula, making it perpetually positive or 0. If entropy is 0, there is no new information to be gained. I will demonstrate the implementation of this formula through an example.

Consider the scenario of a coin toss. There are two probable outcomes, heads or tails, with equal probabilities. If we quantify that, $x \in$ {heads,tails}, and p(heads) = 0.5 and p(tails) = 0.5. If we plug in these values in the formula:

$$H(x) = -\left[\left(p(x_{heads}) * \log\left(p(x_{heads})\right)\right) + \left(p(x_{tails}) * \log\left(p(x_{tails})\right)\right)\right]$$
$$= -[(0.5 * log_2 0.5) + (0.5 * log_2 0.5)] = -[(-0.5) + (-0.5)] = 1$$

Calculating Entropy in a coin toss event

Therefore, entropy is 1 bit, i.e., the coin toss's outcome can be expressed completely in 1 bit. So, to intuitively express Shannon entropy's concept, it is understood as "how long does a message need to be to convey its value completely". I want to dive a little deeper and discuss the concepts of Joint Entropy, Conditional Entropy and Relative Entropy.

Joint and Conditional Entropy

Previously, I defined Entropy for a single random variable, but now I will extend it to a pair of random variables. It is a simple aggregation as we can define the pair of variables (*X*, *Y*) as a single vector-valued random variable.

The joint entropy *H*(*X*, *Y*) of a pair of discrete random variables (*X*, *Y*) with a joint distribution *p*(*x*, *y*) is defined as

$$H(X, Y) = -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log p(x, y)$$

Joint Entropy

This can also be represented in terms of expected value.

$$H(X, Y) = -E \log p(X, Y)$$

Joint Entropy (Expected Value form)

Similarly, for conditional entropy, *H*(*Y*|X) is defined as:

$$H(Y|X) = \sum_{x \in \mathcal{X}} p(x) H(Y|X = x)$$

Conditional Entropy

Intuitively, this is the average of the entropy of *Y* given *X* over all possible values of *X*. Considering the fact that (*X*, *Y*) ~ *p*(*x*, *y*), the conditional entropy can also be expressed in terms of expected value.

$$H(Y|X) = -E \log p(Y|X).$$

Conditional Entropy (Expected Value form)

Let's try an example to understand Conditional Entropy better. Consider a study where subjects were asked:

I) if they smoked, drank or didn't do either.

II) if they had any form of cancer

Now, I will represent these questions' response as two different discrete variables belonging to a joint distribution.

Activity	Cancer
Smoking	Yes
Smoking	Yes
Smoking	Yes
Alcohol	No
Neither	Yes
Alcohol	Yes
Alcohol	Yes
Smoking	No
Alcohol	Yes
Neither	No

Data Table

On the left, you can see the data table where 10 subjects answered the questions. We have three different possibilities for the variable Activity (I will call it X). The second column represents whether the subject has/had cancer (variable Y). There are two possibilities here, i.e. Yes or No. Since we are not dealing with continuous variables yet, I have kept these variables discrete. Let's create a probability table that will make the scenario clearer.

	Yes	No
Smoking	$^{3}/_{10}$	$^{1}/_{10}$
Alcohol	$^{3}/_{10}$	$^{1}/_{10}$
Neither	$^{1}/_{10}$	$^{1}/_{10}$

Probability Table for the above example

Next, I will calculate the value of the marginal probability $p(x)$ for all the possible value of X.

$$p(X = \text{"Smoking"}) = \frac{4}{10}$$

$$p(X = \text{"}Alcohol\text{"}) = \frac{4}{10}$$

$$p(X = \text{"}Neither\text{"}) = \frac{2}{10}$$

Marginal Probability of X

Based on the probability table, we can plug in the value in the conditional entropy formula.

$$
\begin{aligned}
H(Y|X) &= p(\text{"Smoking"})H(Y|X=\text{"Smoking"}) + p(\text{"Alcohol"})H(Y|X=\text{"Alcohol"}) + p(\text{"Neither"})H(Y|X=\text{"Neither"}) \\
&= -\frac{4}{10}\left\{\frac{3}{10}\log\left(\frac{3}{10}\right)+\frac{1}{10}\log\left(\frac{1}{10}\right)\right\} - \frac{4}{10}\left\{\frac{3}{10}\log\left(\frac{3}{10}\right)+\frac{1}{10}\log\left(\frac{1}{10}\right)\right\} - \frac{2}{10}\left\{\frac{1}{10}\log\left(\frac{1}{10}\right)+\frac{1}{10}\log\left(\frac{1}{10}\right)\right\} \\
&= 0.4 * 0.857 + 0.4 * 0.857 + 0.2 * 0.664 \\
&= 0.8184\ bits
\end{aligned}
$$

Conditional Probability in the above example

Relative Entropy

Relative Entropy is somewhat different as it moves on from random variables to distributions. It is a measure of the distance between two distributions. **A more instinctive way to put it would be: Relative entropy or KL-Divergence, denoted by *D*(*p*||*q*), is a measure of the inefficiency of assuming that the distribution is *q* when the true distribution is *p*.** It can be defined as:

$$D(p||q) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}$$

Relative entropy is always non-negative and can be 0 only if $p = q$. Although a point to note here is that it is not a true distance since it is not symmetric in nature. But it is often considered as a "distance" between distributions.

Lets take an example to solidify this concept! Let $X = \{0,1\}$ and consider two distributions p and q on X. Let $p(0) = 1 - r$, $p(1) = r$ and let $q(0) = 1 - s$, $q(1) = s$. Then,

$$D(p||q) = (1-r)\log\left[\frac{1-r}{1-s}\right] + r\log\frac{r}{s}$$

Relative Entropy for p||q

I would also like to demonstrate the non-symmetric property, so I will also calculate $D(q||p)$.

$$D(q||p) = (1-s)\log\left[\frac{1-s}{1-r}\right] + s\log\frac{s}{r}$$

Relative Entropy for q||p

If r = s, $D(p||q) = D(q||p) = 0$. But I will take some different values, for instance, $r = 1/2$ and $s = 1/4$.

$$D(p||q) = \left(1-\frac{1}{2}\right)\log\left[\frac{1-\frac{1}{2}}{1-\frac{1}{4}}\right] + \frac{1}{2}\log\frac{\frac{1}{2}}{\frac{1}{4}} = 0.2075 \text{ bit}$$

$$D(q||p) = \left(1 - \frac{1}{4}\right) \log \left[\frac{1 - \frac{1}{4}}{1 - \frac{1}{2}}\right] + \frac{1}{4} log \frac{\frac{1}{4}}{\frac{1}{2}} = 0.1887 \text{ bit}$$

As you can see, $D(p||q) \neq D(q||p)$.

Now, since we have discussed the different types of entropies, we can move onto Mutual Information.

Mutual Information

Mutual Information is a measure of the amount of information that one random variable contains about another random variable. **Alternatively, it can be defined as the reduction in uncertainty of one variable due to the knowledge of the other.** The technical definition for it would be as follows:

Consider two random variables X and Y with a joint probability mass function $p(x, y)$ and marginal probability mass functions $p(x)$ and $p(y)$. The mutual information $I(X; Y)$ is the relative entropy between the joint distribution and the product distribution $p(x)p(y)$.

$$I(X; Y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}$$
$$= D(p(x, y)||p(x)p(y))$$

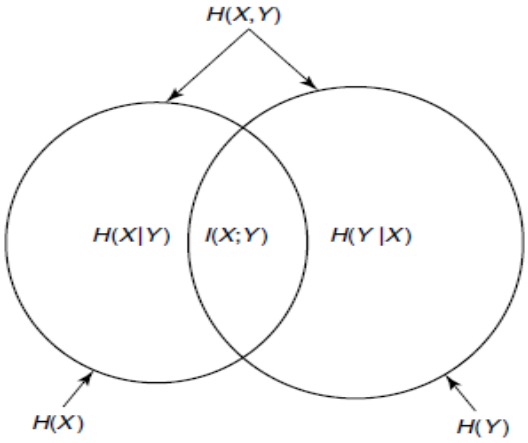
Mutual Information

Mutual Information can also be expressed in terms of Entropy. The derivation is quite fun, but I will refrain myself as it might clutter the article.

$$I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$$

Mutual Information w.r.t. Entropy

From the above equation, we see that mutual information is the reduction in the uncertainty of X due to the knowledge of Y. There is a Venn diagram perfectly describing the relationship.



Relationship between Mutual Information and Entropy

Let's take an example to understand it better. I can use the example of relating Smoking, Drinking, Neither with having cancer that I used while explaining entropy. We had seen that the $H(Y|X) = 0.8184$ bits. To calculate Mutual Information, I require one more term $H(Y)$. $H(Y)$, in this case, will be:

$$\begin{aligned} H(Y) &= -\sum_{y \in Y} p(y) \log p(y) \\ &= \frac{7}{10} \log \frac{7}{10} + \frac{3}{10} \log \frac{3}{10} \\ &= 0.876 \end{aligned}$$

Therefore, Mutual Information is defined by:

$$\begin{aligned} I(X;Y) &= H(Y) - H(Y|X) \\ &= 0.876 - 0.8184 \\ &= 0.0576 \end{aligned}$$

Rassal bir değişkenin belirsizlik ölçütü olarak bilinen Entropi, bir süreç için tüm örnekler tarafından içerilen enformasyonun beklenen değeridir. ***Enformasyon ise rassal bir olayın gerçeklesmesine ilişkin bir bilgi ölçütüdür.*** Eşit olasılıklı durumlar yüksek belirsizliği temsil eder. Shannon'a göre bir sistemdeki durum değiştiğinde entropideki değişim kazanılan enformasyonu tanımlar. Buna göre maksimum belirsizlik durumundaki değişim muhtemelen maksimum enformasyonu sağlayacaktır. Shannon bilgiyi bitlerle temsil ettiği için logartimayı iki tabanında kullanımıştır.

$$I(x) = \log_2 \frac{1}{P(x)} = -\log_2 P(x)$$

Shannon'a göre entropi, iletilen bir mesajın tasıdığı enformasyonun beklenen değeridir.

Shannon Entropisi (H) adıyla anılan terim, tüm x_i durumlarına ait $P(x_i)$ olasılıklarına bağlı bir değerdir.

$$H(X) = E(I(X)) = \sum_{1 \le i \le n} P(x_i).I(x_i) = \sum_{i=1}^{n} P(x_i) \log_2 \frac{1}{P(x_i)} = -\sum_{i=1}^{n} P_i \log_2 P_i$$

$$Log_2(P) = \frac{10}{3} Log_{10}(P)$$

$$H(X) = -\frac{10}{3} \sum_{i=1}^{n} P_i \, Log_{10} P_i$$

10 tabanında logaritmik ifadeler:

- Log1=0, Log 2 ≈ 0.3, Log 3 ≈ 0.477, Log 5 ≈ 0.7, Log 7 ≈ 0.845, Log10=1
- $\log(a*b)=\log a + \log b$; $\log a^n = n*\log a$

Örnek:

Bir paranın havaya atılması olayı, rassal X sürecini temsil etsin. Yazı (1/2) ve tura (1/2) gelme olasılıkları eşit olduğu için X sürecinin entropisi 1 olan para atma olayı (X) gerçeklestiğinde 1 bitlik bilgi kazanılacaktır.

$$H(X) = -\sum_{i=1}^{2} p_i \log_2 p_i = -(0.5 \log_2 0.5 + 0.5 \log_2 0.5) = 1$$

Örnek:

Hava, nem, rüzgar, su sıcaklığı gibi değerlere göre pikniğe gidip gitmeme kararı verilmiş 4 olay sonucunda pikniğe gidildi mi? sorusunun iki yanıtı var. Evet yanıtının olasılığı: ¾, hayır yanıtının olasılığı:1/4.

Olay No	Hava	Nem	Rüzgar	Su sıcaklığı	Pikniğe gidildi mi?
1	güneşli	normal	güçlü	ılık	Evet
2	güneşli	yüksek	güçlü	ılık	Evet
3	yağmurlu	yüksek	güçlü	ılık	Hayır
4	güneşli	yüksek	güçlü	soğuk	Evet

$$H(X) = -\sum_{i=1}^{2} p_i \log_2 p_i$$

Entropy(S)=H(x)=-(3/4)$\log_2$(3/4)-(1/4)$\log_2$(1/4)=0.811

Örnek:

Makine öğrenmesi algoritmasının olasılık hesaplanması sonucunda karar vermesi gerekmektedir. İki durum söz konusudur. Birinci durumun olma olasılığı, P1=0.6, olmama olasığı, P2=0.4 ise entropisini hesaplayınız.

$Log_2(0.6) = -0.743$

$Log_2(0.4) = -4/3$

Entropi, H(x)=0.6*0.743+0.4*4/3=0.979

Calculation of Mutual Information

Alternatively, I can also use *H*(*X*) and *H*(*X*|*Y*) to calculate Mutual Information, and it will yield the same result. We can see how knowing X means so little for the uncertainty of variable Y. Let me change the passage of this example and lead you to how it all makes sense in Machine Learning. Suppose X is a predictor variable and Y is the predicted variable. Mutual Information between them can be a great precursor to check how useful the feature will be for predictions. Let us discuss the implications of Information Theory in Machine Learning.

Applications of Information Theory in Machine Learning

There are quite a few applications around, but I will stick with a few popular ones.

Decision Trees

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. The core algorithm used here is called ID3, which was developed by Ross Quinlan. It employs a top-down greedy search approach and involves partitioning the data into subsets with homogeneous data. **The ID3 algorithms decide the partition by calculating the homogeneity of the sample using entropy.** If the sample is homogenous, entropy is 0, and if the sample is uniformly divided, it has maximum entropy. But entropy does not have direct implication on the construction of the trees. The algorithm relies on Information Gain, which is based on the decrease in entropy after a dataset is split on an attribute. If you think intuitively, you will see that this is actually Mutual Information that I had mentioned above. Mutual Information decreases the uncertainty of one variable given the value of the other. In DT, we calculate the entropy of the predicted variable. Then, the dataset is split based on entropy, and the entropy of the resultant variable is subtracted from the previous entropy value. This is Information Gain and, obviously, Mutual Information in play.

Cross-Entropy

Cross entropy is a concept very similar to Relative Entropy. Relative entropy is when a random variable compares true distribution *p* with how the approximated distribution *q* differs from *p* at each sample point (divergence or difference). Whereas cross-entropy directly compares true distribution *p* with approximated distribution *q*. Now, cross-entropy is a term heavily used in the field of deep learning. **It is used as a loss function that measures the performance of a classification model whose output is a probability value between 0 and 1.** Cross-entropy loss increases as the predicted probability diverge from the actual label.

KL-Divergence

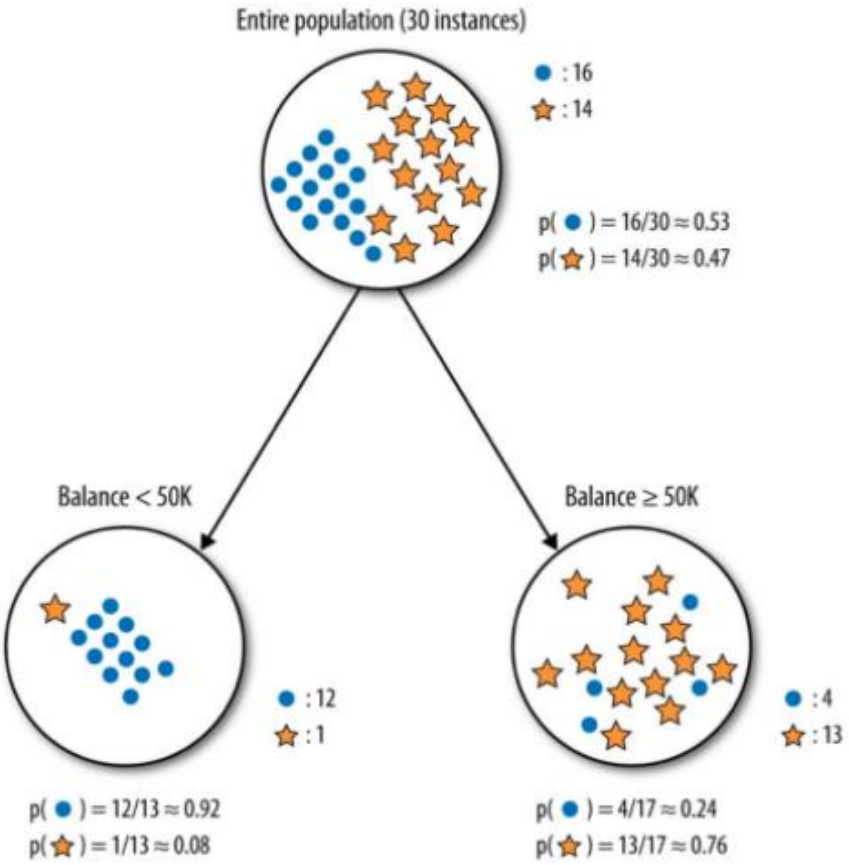
K-L Divergence or Relative Entropy is also a topic embedded in the deep learning literature, specifically in VAE. Variational Autoencoders take in input in the form of Gaussian Distributions rather than discrete data points. It is optimal for the distributions of the VAE to be regularized to increase the amount of overlap within the latent space. **K-L divergence measures this and is added to the loss function.**

K-L Divergence is also used in t-SNE. tSNE is a dimensionality reduction technique that is mainly used to visualize data in high dimensions. It converts similarities between data points to joint probabilities and **tries to minimize the Kullback-Leibler divergence between the joint probabilities of the low-dimensional embedding and the high-dimensional data.**
Calculating imbalance in target class distribution

Entropy can be used to calculate target class imbalances. If we consider the predicted feature as a random variable with two classes, a balanced set (50/50 split) should have the maximum entropy as we saw in the case of the coin toss. But if the split is skewed and one class has a 90% prevalence, then there's lesser knowledge to be gained, hence a lower entropy. Implementing the chain rule for calculating entropy, we can check whether a multiclass target variable is balanced in a single quantified value, albeit an average that masks the individual probabilities.

Örnek: Karar Ağacı

Bir kişiye verilen bir kredinin bir zararla sonuçlanıp sonuçlanmayacağını tahmin etmek için bir karar ağacı oluşturulmuştur. Tüm veri kümesi 30 örnekten oluşmaktadır. 16'sı silinen sınıfa, diğer 14'ü silinmeyen sınıfa aittir. İki değer alabilen "Bakiye" -> "< 50K" veya ">50K" ve üç değer alabilen "Konut" -> "KENDİ", "KİRALIK" veya "DİĞER" olmak üzere iki özelliğimiz var. Entropi ve Bilgi Kazanımı kavramlarını kullanarak bir karar ağacı algoritmasının hangi özniteliğin ilk olarak bölüneceğine ve hangi özelliğin daha fazla bilgi sağladığına veya hedef değişkenimiz hakkındaki belirsizliği ikisinden daha fazla azalttığına nasıl karar vereceğini göstereceğim.



Özellik 1: Denge

Noktalar, sınıf hakkı olan veri noktalarıdır ve yıldızlar, silinmeyenlerdir. Ana düğümü öznitelik dengesine bölmek bize 2 alt düğüm verir. Sol düğüm, silme sınıfından 12/13 (0.92 olasılık) gözlem ile toplam gözlemlerin 13'ünü ve sınıfın yazılmayan sınıfından sadece 1/13 (0.08 olasılık) gözlemi alır. Sağ düğüm, silinmeyen sınıftan 13/17(0.76 olasılık) ve silinen sınıftan 4/17 (0.24 olasılık) ile toplam gözlemin 17'sini alır.

Ana düğümün entropisini hesaplayalım ve Denge üzerinde bölerek ağacın ne kadar belirsizliği azaltabileceğini görelim.

$$E(Parent) = -\frac{16}{30}\log_2\left(\frac{16}{30}\right) - \frac{14}{30}\log_2\left(\frac{14}{30}\right) \approx 0.99$$

$$E(Balance < 50K) = -\frac{12}{13}\log_2\left(\frac{12}{13}\right) - \frac{1}{13}\log_2\left(\frac{1}{13}\right) \approx 0.39$$

$$E(Balance > 50K) = -\frac{4}{17}\log_2\left(\frac{4}{17}\right) - \frac{13}{17}\log_2\left(\frac{13}{17}\right) \approx 0.79$$

Weighted Average of entropy for each node:

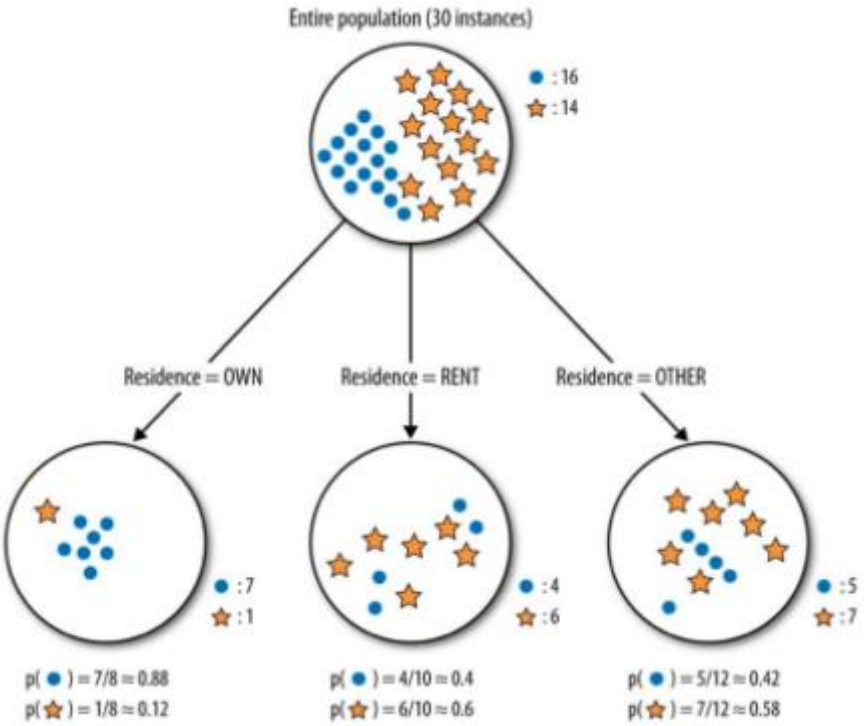
$$E(Balance) = \frac{13}{30} \times 0.39 + \frac{17}{30} \times 0.79$$
$$= 0.62$$

Information Gain:

$$IG(Parent, Balance) = E(Parent) - E(Balance)$$
$$= 0.99 - 0.62$$
$$= 0.37$$

Özelliğe göre bölme, “Denge” hedef değişkenimiz üzerinde 0.37'lik bir bilgi kazanımına yol açar. Aynı şeyi nasıl karşılaştırdığını görmek için “Konut” özelliği için yapalım.

Özellik 2: Konut



Ağacı Residence'ta bölmek bize 3 alt düğüm verir. Sol alt düğüm, silinen sınıftan 7/8 (0.88 olasılık) gözlem ile toplam gözlemlerin 8'ini ve silinmeyen sınıftan sadece 1/8 (0.12 olasılık) gözlem alır. Orta alt düğümler, silinen sınıftan 4/10 (0,4 olasılık) ve silinmeyen sınıftan 6/10(0,6 olasılık) gözlem ile toplam gözlemlerin 10'unu alır. Sağ alt düğüm, silme sınıfından 5/12 (0.42 olasılık) gözlem ve silinmeyen sınıftan 7/12 (0.58) gözlem ile toplam gözlemlerin 12'sini alır. Ana düğümün entropisini zaten biliyoruz. “Konut”tan elde edilen bilgi kazancını hesaplamak için bölmeden sonraki entropiyi hesaplamamız yeterlidir.

$$E(Residence = OWN) = -\frac{7}{8}\log_2\left(\frac{7}{8}\right) - \frac{1}{8}\log_2\left(\frac{1}{8}\right) \approx 0.54$$

$$E(Residence = RENT) = -\frac{4}{10}\log_2\left(\frac{4}{10}\right) - \frac{6}{10}\log_2\left(\frac{6}{10}\right) \approx 0.97$$

$$E(Residence = OTHER) = -\frac{5}{12}\log_2\left(\frac{5}{12}\right) - \frac{7}{12}\log_2\left(\frac{7}{12}\right) \approx 0.98$$

Weighted Average of entropies for each node:

$$E(Residence) = \frac{8}{30} \times 0.54 + \frac{10}{30} \times 0.97 + \frac{12}{30} \times 0.98 = 0.86$$

Weighted Average of entropies for each node:

$$E(Residence) = \frac{8}{30} \times 0.54 + \frac{10}{30} \times 0.97 + \frac{12}{30} \times 0.98 = 0.86$$

Information Gain:

$$\begin{aligned} IG(Parent, Residence) &= E(Parent) - E(Residence) \\ &= 0.99 - 0.86 \\ &= 0.13 \end{aligned}$$

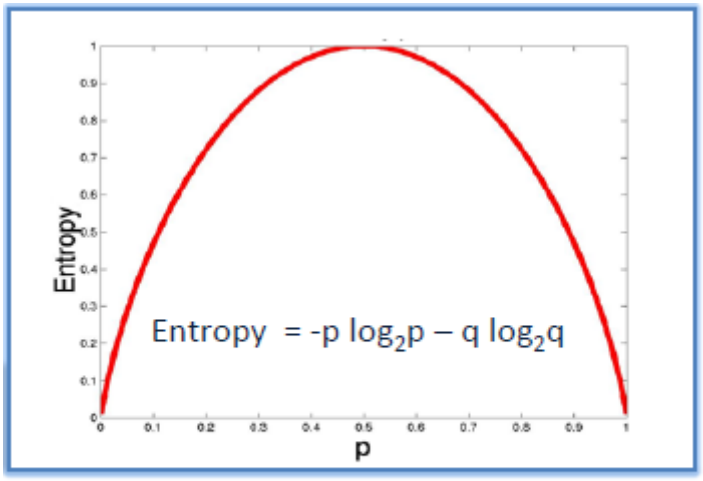
Denge özelliğinden gelen bilgi kazanımı, Konut'tan elde edilen bilgiden neredeyse 3 kat daha fazla! Geri dönüp grafiklere bir göz atarsanız, Denge'de bölünen alt düğümlerin, Yerleşim düğümlerinden daha saf göründüğünü görebilirsiniz. Bununla birlikte, ikamet için en soldaki düğüm de çok saftır, ancak ağırlıklı ortalamaların devreye girdiği yer burasıdır. Bu düğüm çok saf olmasına rağmen, toplam gözlemlerin en az miktarına sahiptir ve bir sonuç, Yığındaki bölmeden toplam entropiyi hesapladığımızda saflığının küçük bir kısmına katkıda bulunur. Bu önemlidir çünkü bir özelliğin genel bilgi gücünü arıyoruz ve sonuçlarımızın bir özellikteki nadir bir değer tarafından çarpıtılmasını istemiyoruz.

Kendi başına Balance özelliği, hedef değişkenimiz hakkında Konuttan daha fazla bilgi sağlar. Hedef değişkenimizde daha fazla düzensizliği azaltır. Bir karar ağacı algoritması, Balance kullanarak verilerimizde ilk bölmeyi yapmak için bu sonucu kullanır. Bundan sonra, karar ağacı algoritması, bir sonraki hangi özelliği böleceğine karar vermek için her bölmede bu süreci kullanacaktır. Gerçek dünya senaryosunda, ikiden fazla özellik ile ilk bölme en bilgilendirici özellik üzerinde yapılır ve daha sonra her bölmede, her bir ek özellik için bilgi kazancının yeniden hesaplanması gerekir, çünkü her birinden elde edilen bilgi kazancı ile aynı olmaz. özellik kendi başına. Entropi ve bilgi kazancı, sonuçları değiştirecek bir veya daha fazla bölme yapıldıktan sonra hesaplanmalıdır. Bir karar ağacı, önceden tanımlanmış bir derinliğe ulaşana ya da hiçbir ek bölünme, genellikle hiper parametre olarak da belirlenebilen belirli bir eşiğin ötesinde daha yüksek bir bilgi kazanımına neden olana kadar derinleştikçe ve derinleştikçe bu işlemi tekrarlar!

Example: Decision Tree - Classification

Outlook	Temp	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No

Bir karar ağacı, bir kök düğümden yukarıdan aşağıya oluşturulur ve verileri benzer değerlere sahip (homojen) örnekler içeren alt kümelere ayırmayı içerir. ID3 algoritması, bir örneğin homojenliğini hesaplamak için entropiyi kullanır. Örnek tamamen homojen (Olma olasılığı 1 ise olmma olasılığı sıfırdır. Tüm olasıklar toplamı bire eşit olmak zorundadır) ise entropi sıfırdır ve örnek eşit olarak bölünmüşse entropisi birdir.



$$\text{Entropy} = -0.5 \log_2 0.5 - 0.5 \log_2 0.5 = 1$$

y=Log2(x)=Log10(x)/0.3

Log10(2)=0.3, Log10(3)=0.477, Log10(5)=0.7, Log10(7)=0.845

Bir karar ağacı oluşturmak için aşağıdaki gibi sıklık tablolarını kullanarak iki tür entropi hesaplamamız gerekir:

a) Bir özelliğin sıklık tablosunu kullanan entropi:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

5/14=0.36
9/14=0.64

b) İki özelliğin sıklık tablosunu kullanan entropi:

$$E(T, X) = \sum_{c \in X} P(c) E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)*__E__(3,2) + **P**(Overcast)*__E__(4,0) + **P**(Rainy)*__E__(2,3)
= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971
= 0.693

Information Gain

Bilgi kazancı, bir veri kümesi bir özniteliğe bölündükten sonra entropideki azalmaya dayanır. Bir karar ağacı oluşturmak, en yüksek bilgi kazancını (yani en homojen dalları) döndüren özniteliği bulmakla ilgilidir.

Adım 1: Hedefin entropisini hesaplanır.

Entropy(PlayGolf) = Entropy (5,9)

= Entropy (0.36, 0.64)

= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)

= 0.94

Adım 2: Veri kümesi daha sonra farklı niteliklere bölünür. Her dal için entropi hesaplanır. Daha sonra, bölme için toplam entropi elde etmek için orantılı olarak eklenir. Ortaya çıkan entropi, bölünmeden önceki entropiden çıkarılır. Sonuç, Bilgi Kazanımı veya entropideki azalmadır.

		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			

		Play Golf	
		Yes	No
Temp.	Hot	2	2
	Mild	4	2
	Cool	3	1
Gain = 0.029			

		Play Golf	
		Yes	No
Humidity	High	3	4
	Normal	6	1
Gain = 0.152			

		Play Golf	
		Yes	No
Windy	False	6	2
	True	3	3
Gain = 0.048			

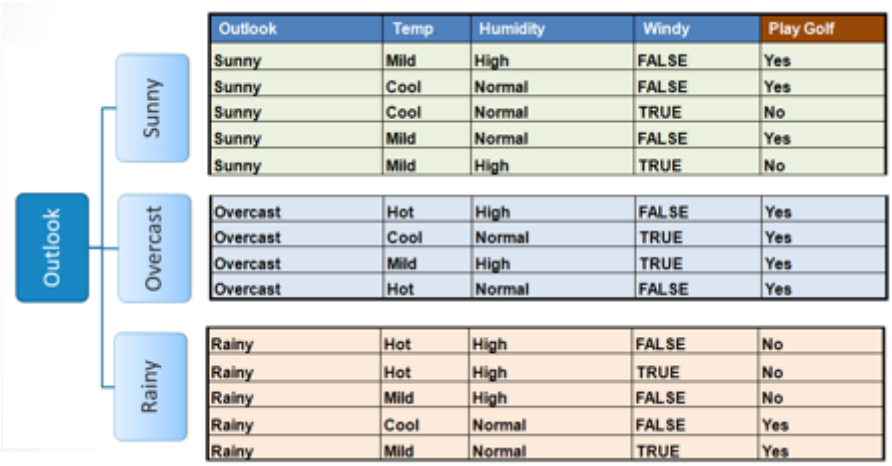
$$Gain(T, X) = Entropy(T) - Entropy(T, X)$$

G(PlayGolf, Outlook) = **E**(PlayGolf) – **E**(PlayGolf, Outlook)

= 0.940 – 0.693 = 0.247

Adım 3: Karar düğümü olarak en büyük bilgi kazancına sahip öznitelik seçilir, veri seti dallarına bölünür ve aynı işlemi her dalda tekrarlanır.

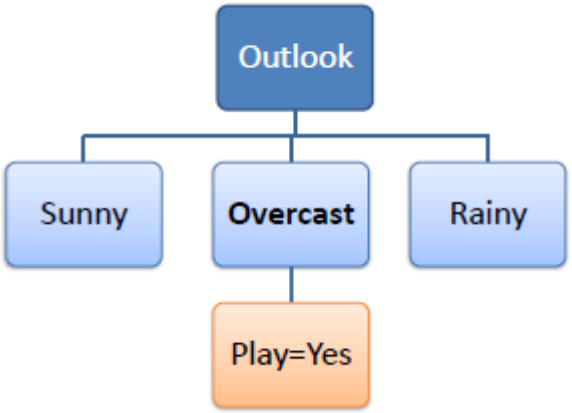
★		Play Golf	
		Yes	No
Outlook	Sunny	3	2
	Overcast	4	0
	Rainy	2	3
Gain = 0.247			



Outlook	Temp	Humidity	Windy	Play Golf
Sunny	Mild	High	FALSE	Yes
Sunny	Cool	Normal	FALSE	Yes
Sunny	Cool	Normal	TRUE	No
Sunny	Mild	Normal	FALSE	Yes
Sunny	Mild	High	TRUE	No
Overcast	Hot	High	FALSE	Yes
Overcast	Cool	Normal	TRUE	Yes
Overcast	Mild	High	TRUE	Yes
Overcast	Hot	Normal	FALSE	Yes
Rainy	Hot	High	FALSE	No
Rainy	Hot	High	TRUE	No
Rainy	Mild	High	FALSE	No
Rainy	Cool	Normal	FALSE	Yes
Rainy	Mild	Normal	TRUE	Yes

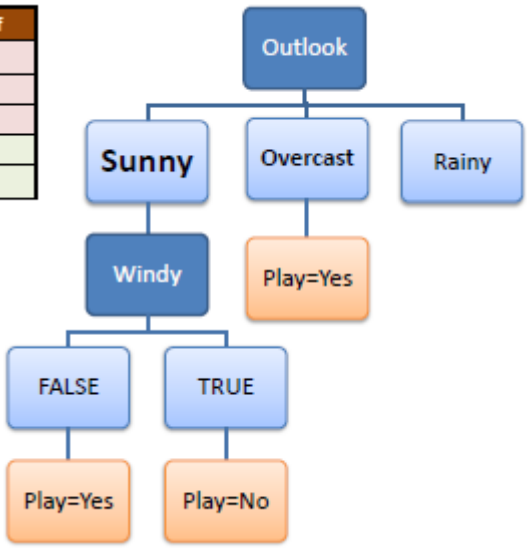
Step 4a: Entropisi 0 olan bir dal, bir yaprak düğümdür.

Temp	Humidity	Windy	Play Golf
Hot	High	FALSE	Yes
Cool	Normal	TRUE	Yes
Mild	High	TRUE	Yes
Hot	Normal	FALSE	Yes



Adım 4b: Entropisi 0'dan büyük olan bir dalın daha fazla bölünmesi gerekir.

Temp	Humidity	Windy	Play Golf
Mild	High	FALSE	Yes
Cool	Normal	FALSE	Yes
Mild	Normal	FALSE	Yes
Cool	Normal	TRUE	No
Mild	High	TRUE	No



Adım 5: ID3 algoritması, tüm veriler sınıflandırılana kadar yaprak olmayan dallarda özyinelemeli olarak çalıştırılır.

Karar Ağacından Karar Kurallarına

Bir karar ağacı, kök düğümden yaprak düğümlere tek tek eşlenerek kolayca bir dizi kurala dönüştürülebilir.

R_1: IF (Outlook=Sunny) AND (Windy=FALSE) THEN Play=Yes

R_2: IF (Outlook=Sunny) AND (Windy=TRUE) THEN Play=No

R_3: IF (Outlook=Overcast) THEN Play=Yes

R_4: IF (Outlook=Rainy) AND (Humidity=High) THEN Play=No

R_5: IF (Outlook=Rain) AND (Humidity=Normal) THEN Play=Yes