Model Evaluation Metrics

1) Confusion Matrix:

		Actual Values: Positive	Actual Values: Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

True or False   Prediction

TP → We predict Positive and it's TRUE.

TN → We predict Negative and it's TRUE

(Type 1 error) FP → We predict Positive and it's False

(Type 2 error) FN → We predict Negative and it's False.

- Recall or Sensitivity = Pozitif class'ların ne kadarını doğru sınıflandırdık. $\frac{TP}{TP+FN}$

- Precision = $\frac{TP}{TP+FP}$ ⇒ Pozitif predict ettiklerimizin ne kadarı doğru.

- Specifity = Negatif class'ların ne kadarını doğru sınıflandırdık.

$$\frac{TN}{TN+FP}$$

• Accuracy = $\dfrac{TP+TN}{TP+TN+FP+FN}$

Not : 1) Algorithms like SVM and KNN create a class output. Yani, outputs will be either 0 or 1. Bu algoritmalar class outputs(1,0) çıkarır; probability çıkarmaz.
2) Logistic Reg., Random Forest, Boosting etc. give probability outputs. Converting probability outputs to class output is just a matter of creating a threshold probability.

• F1 Score = f1 skor'a girmeden önce; Recall ve Precision arasındaki Trade-off'a bakalım.

Trading off Precision and Recall:

ÖR, Logistic Regression $0 \leq h_\theta(x) \leq 1$ (olasılık)

Predict 1 if $h_\theta(x) \geq$ ~~0.5~~ 0.7

Predict 0 if $h_\theta(x) <$ ~~0.5~~ 0.7

İlk durumda cutoff değerimiz 0.5; ancak we want to predict y=1 (cancer) only if very confident. Bu durumda cut-off değerimizi 0.7 aldığımızı varsayalım. Bu ne demek? "Tell someone that they have cancer only if we think greater than or equal to 70% chance that they have cancer" yani kişinin kanser olma ihtimali 0.7 ve üzeri ise kişiye kanserli diyoruz, aksi durumda (%70'in altındaki olasılıkta) kanserli değil (y=0) diyoruz.

Bu durumda;

↳ Higher Precision, Lower Recall.

Çünkü; FP'yi azalttık, ancak FN arttı.

(Threshold'u artırarak Pozitif tahmin ettiğimiz, ama gerçekte negatif olan gözlemleri azaltmış olduk.)

../..../....

Tersi durumda; Lower threshold probability (0.3) durumunda FN değerlerini azaltmış oluruz.

Bu durumda;

↳ Higher Recall, lower Precision.

Gelelim F1 skor'a:

$$F_1 = 2 \cdot \frac{Precision \times Recall}{Precision + Recall}$$

* F1 skoru, is a better metric when there are imbalanced classes. (Eşit dağılmayan veri kümelerinde hatalı bir model seçimi yapmamızı engeller)

* Precision ve Recall değerlerinin herikisinin de problem açısından önem taşıdığını düşünüyorsak f1 skor genel Model başarısı ölçmek için kullanılır.

FBeta Skor: Bazen, FP'nin en aza indirilmesinin daha önemli olduğu, ancak FN'lerin hâlâ önemli olduğu durumlarda veya tam tersi durumlarda FBeta ölçümü ile ilgileniriz.

F1 skor genel model başarısı ölçümünde kullanılıyordu ve Precision, Recall değerlerinin her ikisi de hesaplanmasında etkili idi. FBeta skor birinin daha etkili ve önemli olduğu durumda kullanılır.

$$F_\beta = (1+\beta^2) \times \frac{Precision \times Recall}{(\beta^2 \times Precision) + Recall}$$

Örneğin (beta = 0.5) → $F_{0.5}$ Measure ⇒ More weight on Precision, less weight on Recall.

(beta = 1) → F1 skor ⇒ Balance weight

(beta = 2) → F_2 Measure ⇒ less weight on Precision, more weight on Recall.

→ Recall'un Precision'dan 2 kat daha önemli olduğunu düşündüğümüz durum.

$$F_1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP+FN)}$$

$$F_\beta = \frac{(1+\beta^2) \cdot TP}{(1+\beta^2) \cdot TP + \beta^2 \cdot FN + FP}$$

- When we care more about minimizing FP than minimizing FN, we would want to select a β value < 1.
- When the priority is to minimize FN, we would select β value > 1.

→ Seçilen β'ya bağlı olarak F skoru en iyi yapan probability threshold değişecektir. (Fbeta skor - Probability Threshold Grafiği çizdirilerek sonuçlar gözlemlenir.) → Ve en iyi skoru veren threshold seçilir.

İspat: The F-measure was derived by Rijsbergen (1979) so that F_β "measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision". It is based on Effectiveness Measure shown by $E = 1 - \frac{1}{\frac{\alpha}{Precision} + \frac{1-\alpha}{Recall}}$; $F_\beta = 1 - E$ where $\alpha = \frac{1}{1+\beta^2}$

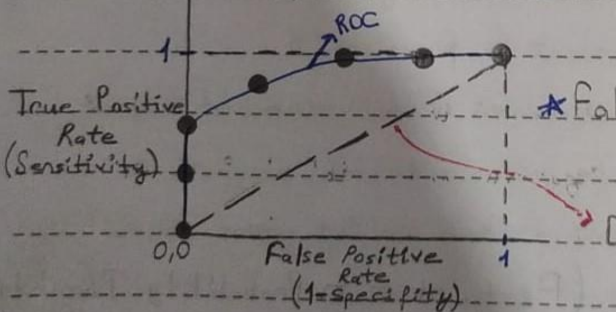
(Effectiveness Function)

The origin of the definition of F Measure.

2) AUC-ROC and Precision-Recall Curve

İlk olarak ROC curve'e bakalım.

Her Threshold değeri için binlerce Confusion Matrix yapılabilir. Tüm thresholdları summarize eden metot "ROC" graph çizdirmektir. Her confusion Matrix'e ait TP rate ve FP rate'leri ROC ile göstermiş oluruz.



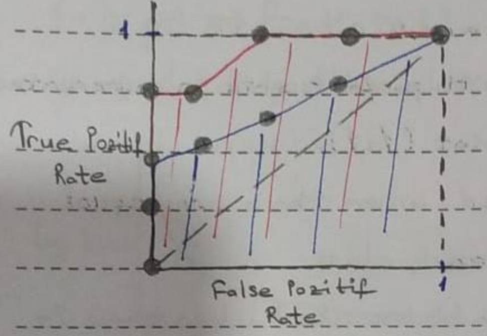
* True Positive Rate $= \frac{TP}{TP+FN}$

* False Positive Rate $= \frac{FP}{FP+TN}$

* ROC graph summarizes all of the confusion Matrixes that each threshold produce.

→ Depending on How many F.P, I am willing to accept, the optimal Threshold is choosen.

AUC Bakalım;



The AUC for the Red ROC curve is greater than the AUC for the Blue ROC curve

Red → Logistic Regression AUC : 0,9

Blue → Random Forest AUC : 0,75

↳ Choose Logistic Regression

* (Diagonalde AUC : 0,5 ⇒ Yani No skill)

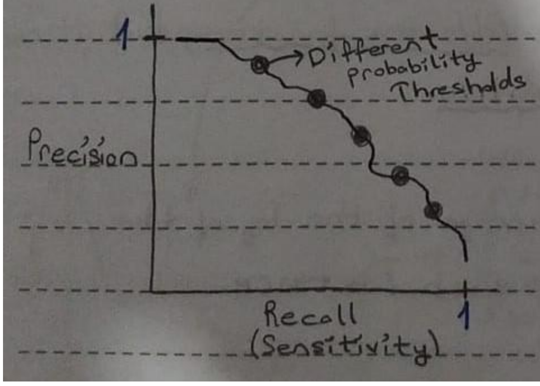
Precision – Recall Curve

(Sensitivity)

* ROC curves are appropriate when the observations are balanced between each class, Precision – Recall curves are appropriate for imbalanced Datasets.

$$\text{Precision} = \frac{TP}{TP+FP}$$

} Does not include the number of True Negatives in its calculation, and is not affected by the imbalance.



→ The curve is created by showing the Precision-Recall for each Threshold value.

Not → AUC yine aynı şekilde kullanılır.

Özet: 1) ROC → Model with perfect skill is represented at a point (0,1).

2) Precision - Recall → Model with perfect skill is represented at a point (1,1).

3) ROC curves should be used when there are equal numbers of observations for each class.

4) Precision - Recall curves should be used when there is class imbalance.

5) ROC curve make it easy to identify the best threshold.

6) AUC can help you decide which classification algorithm (method or model) is better.

3) Log Loss (Binary Cross Entropy)

Tahmindeki olasılık değerlerine dayanan sınıflandırma için önemli bir ölçüttür. Log loss ne kadar düşük olursa, model başarısı o kadar yüksek olur.

log is calculated to base 2 ($\log_2$).

$$* \quad H_p(q) = -\frac{1}{N} \sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1-y_i) \cdot \log(1-p(y_i))$$

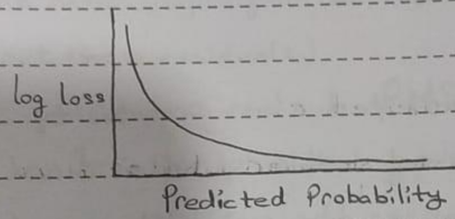
* Log loss is often used as the Objective function, but it can also be used as a performance metric.

* log Loss is just negative average of the log of the corrected probabilities for each instance.

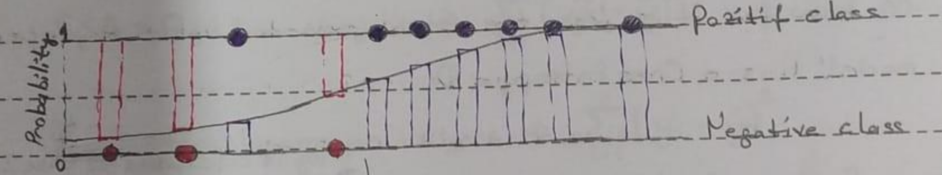
$p(y_i) \Rightarrow$ Predicted probability of positive class.

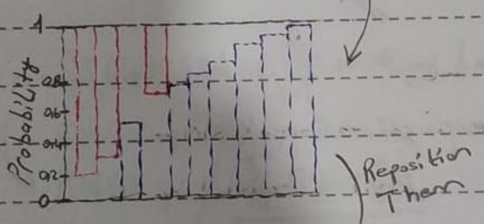
$1 - p(y_i) \Rightarrow$ Predicted " of negative "

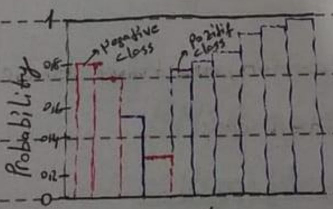
$y_i \Rightarrow$ 1 for positive class and 0 for negative class



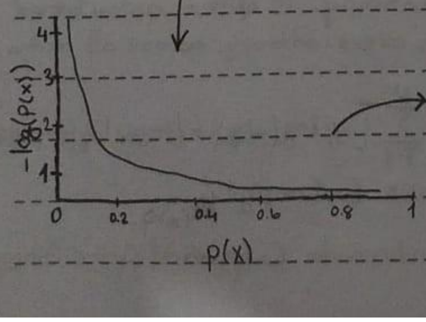
$\Longrightarrow$ Lower the log-loss, better the Model.

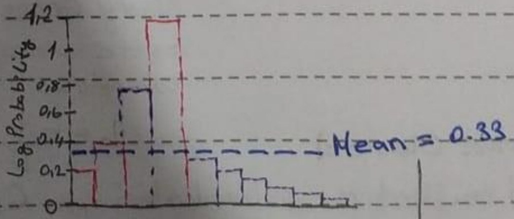






$\rightsquigarrow$ Take the negative log of probability (since the log of values between 0 and 1 is negative, we take negative log to obtain positive value for the loss)





log loss is 0.33 ✓

4) Gini Coefficient

Gini Coefficient = (2 × AUC) − 1

Not:

Cross-Entropy Loss Function → Each predicted class probability is compared to the actual class desired output 0 or 1 and a loss is calculated that penalizes the probability based on how far it is from the actual expected value. A perfect model has a Cross Entropy = 0.

log is calculated to base 2. ($\log_2$)

Cross Entropy is defined as;

$$L_{CE} = -\sum_{i=1}^{n} y_i \log(p(y_i)), \text{ for } n \text{ classes}$$

Cross Entropy Loss (Genel Gösterim)

where y_i → Gerçek sınıf

$p(y_i)$ → Probability

(Log Loss) Binary Cross Entropy;

For binary classification, we have binary cross entropy defined as;

$$L = -\sum_{i=1}^{2} y_i \log(p(y_i))$$

Binary Problem olduğunda yukarıdaki formül (L_{CE}) bu şekilde de yazılabilir

$$= -[y_i \log_2(p(y_i)) + (1-y_i)\log(1-p(y_i))]$$

Binary Cross Entropy is often calculated as the average cross-entropy across all data examples.

$$L = -\frac{1}{N}\left[\sum_{i=1}^{N} [y_i \log(p(y_i)) + (1-y_i)\log(1-p(y_i))]\right]$$

(İsimlendirme Clearet)

* 2'li sınıflandırma problemleri için Cross Entropy'i Binary Cross Entropy (Log Loss); multi-class sınıflandırmada Categorical Cross Entropy isimleri ile adlandırırız.

5) RMSE

Most popular metric for Regression Problems.

RMSE, tahmin hatalarının standard sapmasıdır. RMSE, hataların ne kadar yayıldığının bir ölçüsüdür (RMSE is a measure of how spread out these residuals.)

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (predicted_i - Actual_i)^2}{n}}$$

Gelin bir hatırlatma yapalım:

SSR = Açıklanan Varyans $\Rightarrow \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2$

SSE = Açıklanamayan Varyans $\Rightarrow \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

SST = SSR + SSE = Total varyans in y_i $\Rightarrow \sum_{i=1}^{n} (y_i - \bar{y})^2$

① $RMSE = \sqrt{\frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{n}}$

② $SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$

RMSE can be interpreted as Standard Deviation of the Unexplained Variance. (Tahmin hatalarının standard sapması)

6) Concordant - Discordant Ratio

Ideal bir modelde; Tüm Gerçek 1'lerin olasılık skorları, Tüm gerçek 0'ların olasılık skorlarından daha büyük olmalıdır. Böyle bir modelin mükemmel uyumlu olduğu söylenir.

★ ↳ Modelin ne kadar iyi olduğu konusunda tek başına pek bir şey söylemez. Concordance measure'ü diğer metriklerle kullanmak gerekir.

Örnek ile Anlatalım

farzelim Datamız 4 gözlem içeriyor

Gözlem No	True Class (Actual sınıf)	Probability Score
P_1	1	0.9
P_2	0	0.42
P_3	1	0.30
P_4	1	0.80

1) İlk olarak 1 ve 0'ları içeren tüm pairler yaratılır.

2) Bu örnekte 3 olası pair vardır (1 ve 0 içeren)

$(P_1 - P_2)$, $(P_3 - P_2)$, $(P_4 - P_2)$

3) True 1'in olasılık skoru, True 0'ın olasılık skorundan büyükse bu çifte Concordant (uyumlu) denir.

4) $P_1 - P_2 \Rightarrow 0.9 > 0.42$ } Concordant

$P_3 - P_2 \Rightarrow 0.3 < 0.42$ } Discordant

$P_4 - P_2 \Rightarrow 0.8 > 0.42$ } Concordant

5) Concordance Ratio $= 2/3 = 0.66$

6) Perfect Model Concordance Ratio = 100% >>> Amaç buna erişmek.

In simpler words, we take all possible combinations of Actual 1 and 0. Then, "Concordance" is the percentage of pairs, where Actual 1's probability scores are greater than the scores of Actual 0's.

7) In case both probabilities were equal we call them as tied pairs.

7) Gain and Lift Charts (Curve)

Confusion Matrix can give us a good idea about how effective our model is. But sometimes, we want to know how a particular model does with more data. For example, does a model perform better with %60 of data, compared to 50%? This is where gain and lift charts come in.

Steps to build a Lift/Gain chart:

1) Calculate probability for each observation.
2) Rank these probabilities in decreasing order.
3) Build deciles (genellikle percentile) with each group having almost 10% of the observations.
4) Calculate the target rate at each decile for Target=1, Target=0 and Total.

../..../....

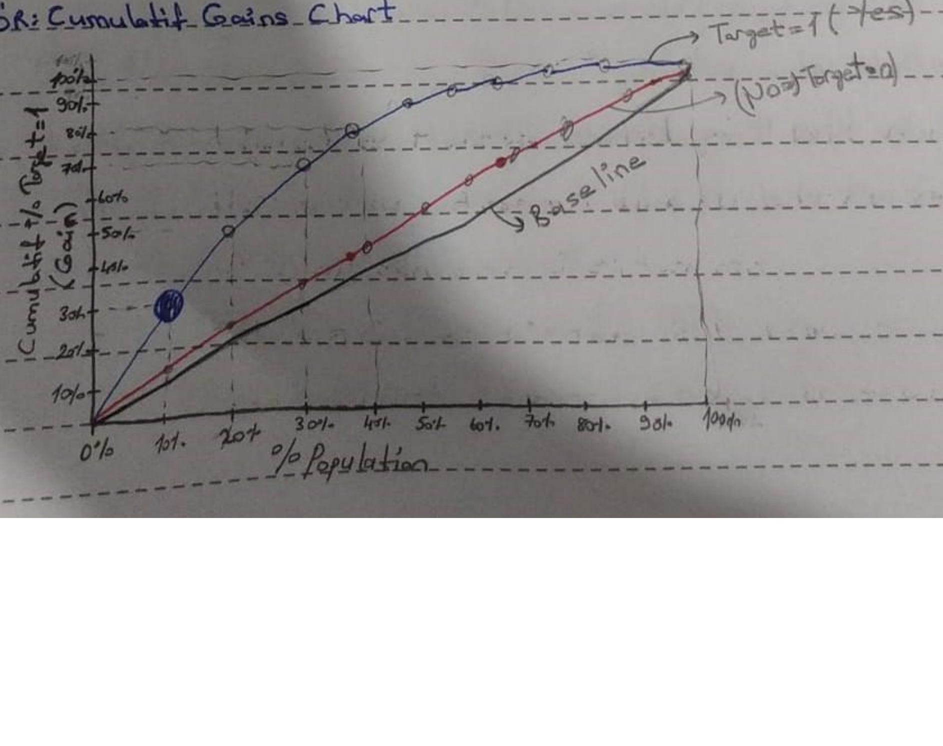
* While the confusion matrix gives proportions between all negatives and positives Gain and Lift charts focus on the $\boxed{\text{True Positives.}}$

Decile	Label (Actual) 0	1	Total	% Target=1	% Target=0	% Population	Cum % Target=1	Cum % Popu
1	0	543	543	14%	0%	10%	14%	%10
2	2	542	544	14%	〃	〃	28%	%20
3	7	537	544	14%	〃	〃	42%	%30
4	15	529	544	14%	1%	〃	56%	%40
5	20	524	544	14%	1%	〃	69%	%50
6	42	502	544	13%	3%	〃	83%	%60
7	104	440	544	11%	7%	〃	94%	%70
8	345	199	544	5%	22%	〃	99%	%80
9	515	29	544	1%	32%	〃	100%	%90
10	540	5	545	0%	34%	〃	100%	%100
Total	1590	3850	5440					

Aslında Herşey Buradan geliyor.

(Örnekteki rakamlar ile grafik rakamları farklı, ancak herşey yukarıdaki tablodan çiziliyor)

1) ÖR: Cumulatif Gains Chart



../..../....

→ for example, Evet (Target=1) (default customer) için eğrinin ilk noktası (%10, %30)'dur. Yani; her gözlem için probability hesaplarsınız ve bu prob'ları büyükten küçüğe sıralarsanız ve; ilk %10 gerçekte Evet (Target=1) alan tüm gözlemlerin %30'unu içerir.

Notes:

I) The diagonal line is the "Baseline" curve; if you select 10% of the cases from the scored dataset at random, you would expect to "gain" 10% of all of the cases that actually take the category of Yes (Target=1).

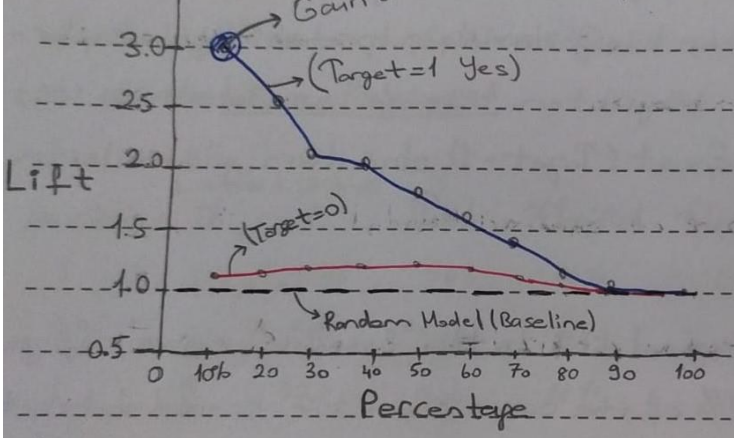
II) The farther above the baseline a curve lies, the greater the gain.

III) We can use the "Cumulative Gains Chart" to help choosing a classification cutoff by choosing a percentage that corresponds to desirable gain, and then mapping that percentage to the appropriate cutoff value.

2) ÖR: Lift Chart

The lift chart is derived from the Cumulative Gains Chart; the values on the y-axis correspond to the ratio of the Cumulative gain for each curve to the baseline. Thus, the lift at 10% for the class Yes(Target=1) is 30% / 10% = 3. it provides another way of looking at the Gain chart.

Gain chart'da %30/ 10% = 3



★ Söylenebilir ki;

For the top 10% predictions, our model is 3x better than random model. For %20 is 2.5x.