MODEL EVALUATION – CLASSIFICATION MODELS

datavedas.com/model-evaluation-classification-models

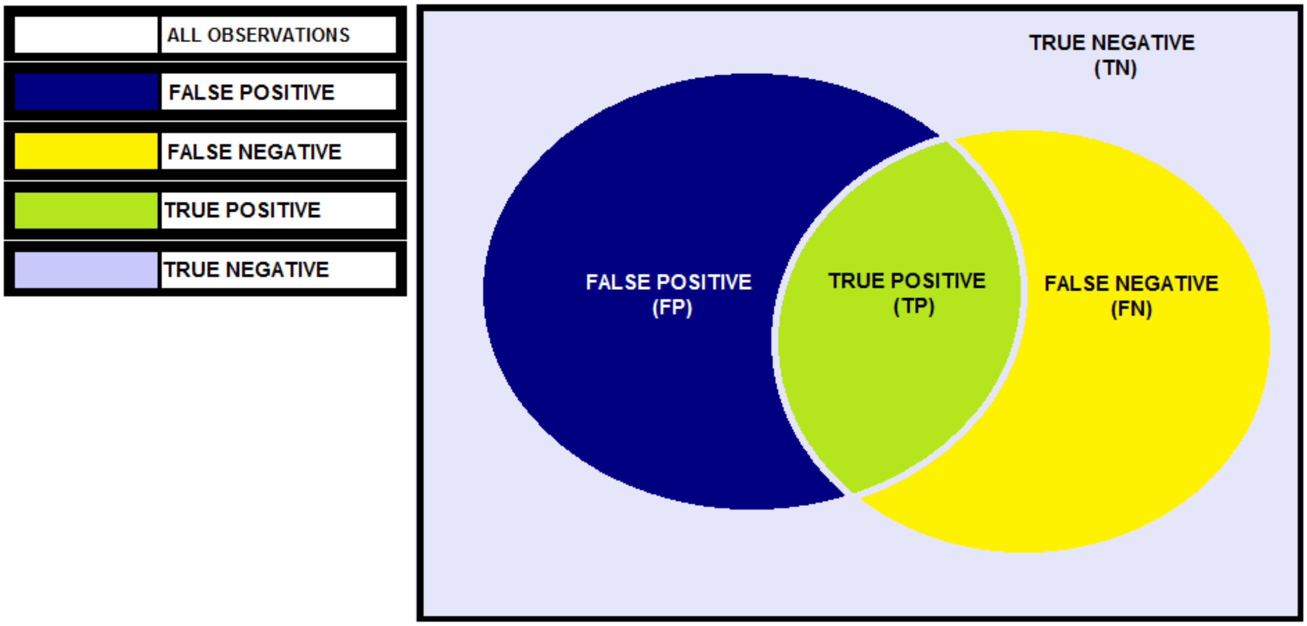
DataVedas

Various model evaluation techniques can be used under the supervised learning setup that helps us in finding the performance of the model. A very simple method to evaluate a model is by finding the accuracy which is the difference between the predicted and the actual values and when we are working with classification models then by accuracy what we mean is the count of the correct predictions of the class labels, however, it is not a perfect method and can lead to poor decision making. Therefore, we need other measures to evaluate the various models and picking the right measure of evaluation is crucial in selecting and distinguishing the right model from other models.

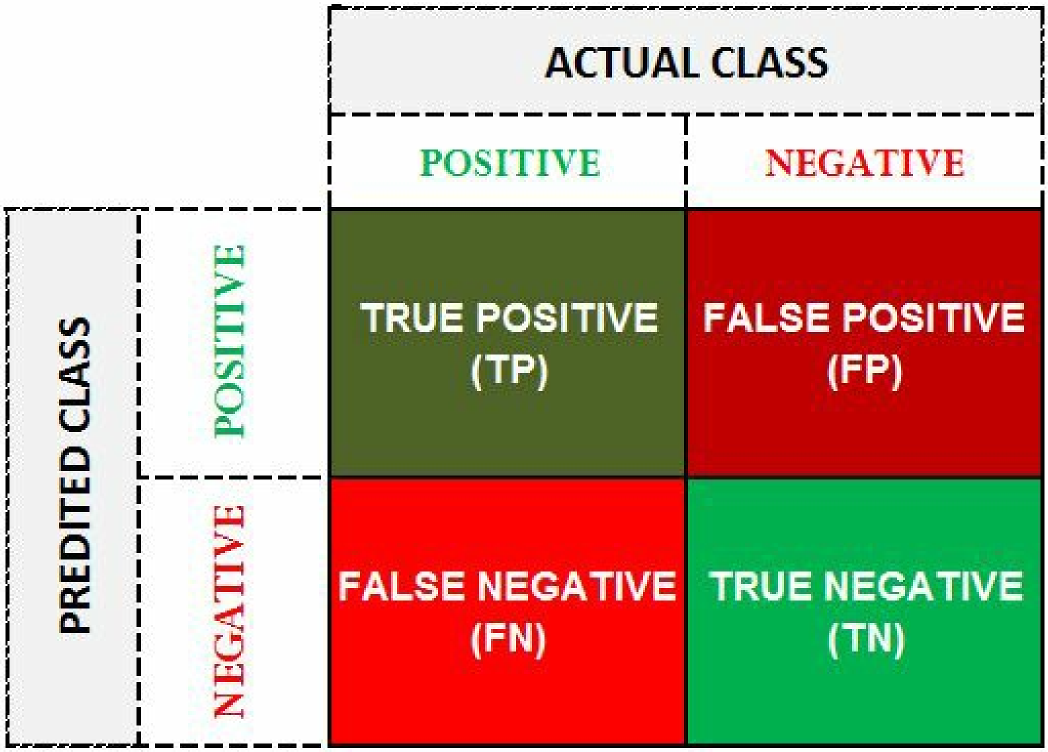
To understand the various metrics that can be used to evaluate a classification problem, we first need to understand what we mean by a Base Rate Model. For example, we have a dataset with 100 records where 90 records belong to class label '0' while the remaining 10 observations belong to class label '1'. As the frequency of the class label '0' is higher than class label '1' then a model which predicts all observations to have the class label '0' will have an accuracy of as high as 90%. Therefore, the accuracy here provides us with little information on how actually the model is performing. It is also important for our model to have accuracy at least more than 90% because a model that simply predicts the most frequent class has the accuracy of 90%. This limitation of accuracy asks for more complex measures of evaluation.

Confusion Matrix

For example, we have a dataset having class labels 0 and 1 where 0 stands for 'Non-Defaulters' while 1 stands for 'Defaulters'. We build a logistic regression model to predict the class label 1. Below is a Venn diagram where all the observations are in the square box. All the observations that were predicted as 1 by the model are represented as the Blue Circle. All the observations that were actually 1 are represented by the yellow circle. The common area of these two circles is denoted by green and contains the observations that were correctly predicted by the model and are called True Positives. False Positive observations are those observations that were incorrectly predicted as 1 by our model while False Negative are those observations that were incorrectly predicted as having the class label '0' by our model. The remaining observations are the True Negative which are those observations that were correctly predicted as having the class label '0' by the logistic regression model.



Rather than a Venn diagram, we can come up with a confusion matrix to show all these observations. For a two-class classification problem, the confusion matrix will have two rows and two columns where the rows represent the actual values whereas the columns represent predicted values. Our objective is to minimize the False Positives and False Negatives and maximise the True Positives and True Negatives thus we want to maximise the diagonal values.



		ACTUAL CLASS	
		POSITIVE	NEGATIVE
PREDITED CLASS	POSITIVE	TRUE POSITIVE (TP)	FALSE POSITIVE (FP)
	NEGATIVE	FALSE NEGATIVE (FN)	TRUE NEGATIVE (TN)

Positive (1) – Defaulter

Negative (0) – Non-Defaulter

True Positive- Model predicted ‘Defaulter’ and were actually ‘Defaulter’

True Negative- Model predicted ‘Non-Defaulter’ and were actually ‘Non-Defaulter’

False Positive- Model predicted ‘Non-Defaulter’ and were actually ‘Defaulter’ (Type I Error)

False Negative- Model predicted ‘Defaulter’ and were actually ‘Non-Defaulter’ (Type II Error)

The real power of confusion matrix lies with the various metrics that can be calculated using the confusion matrix. The most basic measures involve Classification Error and Accuracy.

Classification Error

The formula for Classification Error is (FP + FN) ÷ (TP + TN + FP + FN). Here we divide all the errors with all the observations (ERRORS ÷ TOTAL).

Accuracy

It is simply 1-Error. Here we divide all the correct predictions with all the observations (CORRECT ÷ TOTAL). The formula for Accuracy is (TP + TN) ÷ (TP + TN + FP + FN)

However, as mentioned earlier, these measures can be very misleading. Thus, we come up with more complex measures to evaluate the model. Some of
these measures are mentioned below:

False Positive Rate aka False Alarm rate aka 1-Specificity aka 1-True Negative Rate

It provides us with the percentage of negatives that were classified as positive by the model. Here we divide the false positive with the total number of observations that could have come under False Positive making the formula to be FP ÷ (FP + TN)

False Negative Rate aka Miss rate

It provides us with the percentage of the positives that were classified as negative by the model. Thus, out of all the positives in the data, the positives that were miss-classified by the model is indicated by the False Negative Rate. The formula for it is FN ÷ (FP + TN)

Recall aka Sensitivity aka True Positive Rate aka 1-False Negative Rate

It is the opposite of False Negative Rate. It provides us with the percentage of positives that were classified correctly by the model. The formula for it is = TP ÷ (TP + FN)

Precision aka Positive Predicted Value

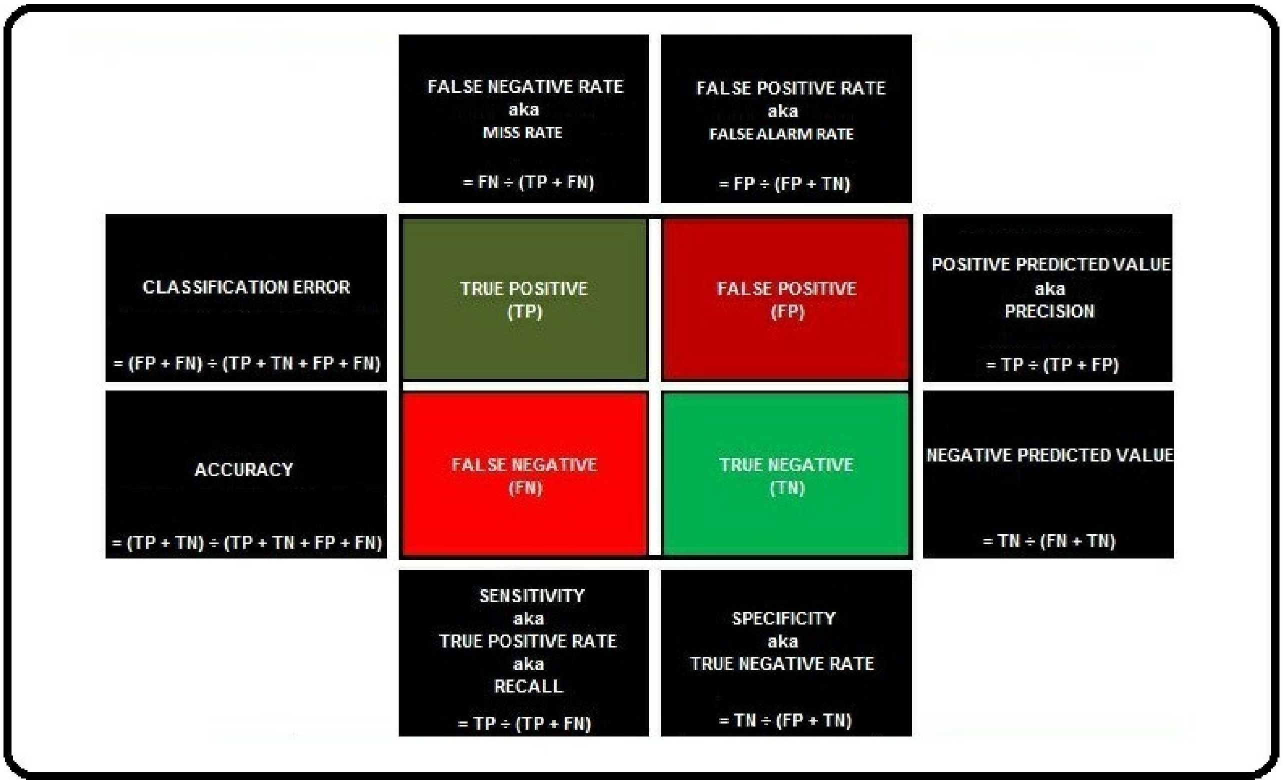
Out of all the predicted positives, the fraction of the actual positive is indicated by precision. Thus, precision provides us with the percentage of the positive out of what the model predicted as positive. Therefore, unlike Recall which represents the percentage of True positive out of all the actual positives, Precision provides us with the percentage of True-positive out of all predicted positives. By precision, we get to know how many of the positives predicted are relevant which help us to evaluate the model. The formula for Precision is = TP ÷ (TP + FP)

Specificity aka True Negative Rate

It indicates how good the model is at avoiding the misclassification of the negatives I.e. Out of the predicted Negatives, how many of them are actually negatives. The formula for Specificity is = TN ÷ (FP + TN)

Negative Predicted Value

Here we divide the True Negatives with the False Negative and True Negative. The formula for Specificity is = TN ÷ (FN + TN)



	FALSE NEGATIVE RATE aka MISS RATE = FN ÷ (TP + FN)	FALSE POSITIVE RATE aka FALSE ALARM RATE = FP ÷ (FP + TN)	
CLASSIFICATION ERROR = (FP + FN) ÷ (TP + TN + FP + FN)	TRUE POSITIVE (TP)	FALSE POSITIVE (FP)	POSITIVE PREDICTED VALUE aka PRECISION = TP ÷ (TP + FP)
ACCURACY = (TP + TN) ÷ (TP + TN + FP + FN)	FALSE NEGATIVE (FN)	TRUE NEGATIVE (TN)	NEGATIVE PREDICTED VALUE = TN ÷ (FN + TN)
	SENSITIVITY aka TRUE POSITIVE RATE aka RECALL = TP ÷ (TP + FN)	SPECIFICITY aka TRUE NEGATIVE RATE = TN ÷ (FP + TN)	

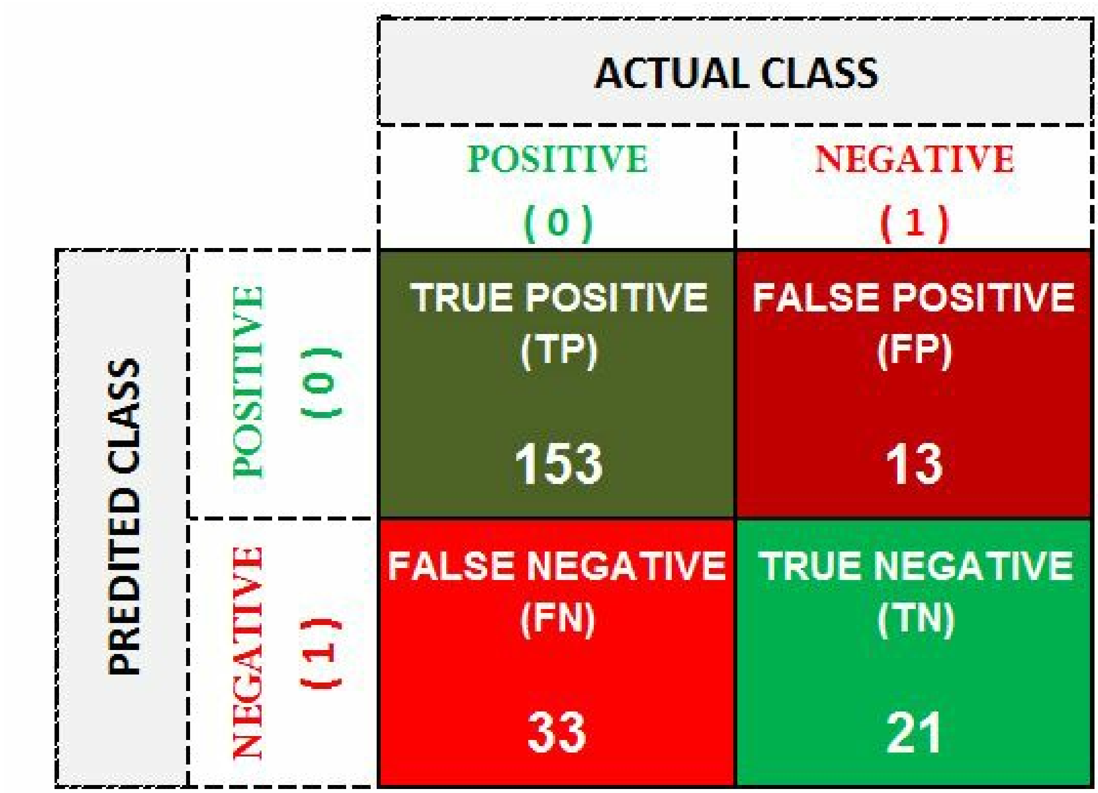
All the above-mentioned metrics are useful, however, they alone cannot provide us with a good evaluation score. For example, if a model predicts all the observations as class label ‘0’ then the False Positive will be nill or a model that predicts class label ‘1’ for all the observations will have a 100% recall. Thus, we need to consider these values in pairs such as Recall-Precision, False Positive-False Negative Rate, True Positive-False Negative Rate etc.

ROC Curve

For understanding the need of ROC Curve, we first need to understand the role played by the threshold in the classification models.

For example, we have a dataset with 220 observations having demographic, financial and other information about people who have taken a loan from XYZ Bank. The observations have two classes, class labels ‘0’ and ‘1’ where ‘0’ denotes Non-Defaulters i.e. those people who were able to repay their loans while ‘1’ indicates ‘defaulters’ i.e. those people who failed to repay their loan.

We fit a logistic regression model on this data to solve this binary classification problem and predict the class label ‘0’ and come up with the following confusion matrix.



		ACTUAL CLASS	
		POSITIVE (0)	NEGATIVE (1)
PREDITED CLASS	POSITIVE (0)	TRUE POSITIVE (TP) 153	FALSE POSITIVE (FP) 13
	NEGATIVE (1)	FALSE NEGATIVE (FN) 33	TRUE NEGATIVE (TN) 21

With the following confusion matrix, we know that the actual number of non-defaulters is 186 (TP + FN) while the actual number of defaulters is 33 (FP + TN). The Total Number of Predicted Non-Defaulters is 165 (TP + FP) while the Predicted Defaulters are 54 (FP + TN). Now if we concentrate on the predicted negatives then we can calculate specificity which tells us out of the predicted Negatives, how many of them are actually negatives and from the above confusion matrix we get specificity to be at 64% (the formula for Specificity is TN ÷ (FP + TN)).

This means that as far as the defaulters are concerned we are 64% accurate in classifying the defaulters. However, many of the classification techniques such as Logistic Regression, Decision Trees, Naive Bayes doesn't explicitly provide us with class labels rather they compute the probability of an observation belonging to a certain class. By default this threshold is set at 50% which means in our case any observation that has the probability of less than 50% will be provided with the class label '1' i.e. will be labelled as a defaulter. As mentioned above, the various measures discussed under confusion matrix can be easily manipulated as a model predicting all observations as '1' will increase the Specificity, thus, we will be able to classify all the defaulters correctly, however, by doing so we will end up labeling a lot of non-defaulters as defaulters and we will end up increasing the False Negatives. This decision of how much of False Negative or False Positive is admissible is very domain specific. For example, if the Bank XYZ is liberal and can tolerate bad debts and wants to grant the loan to more people then the bank will want to have high Sensitivity and will tolerate low Specificity. However, if the bank is stringent and wants to be very cautious in dispersing loans then it won't bother low Sensitivity and will want to have high

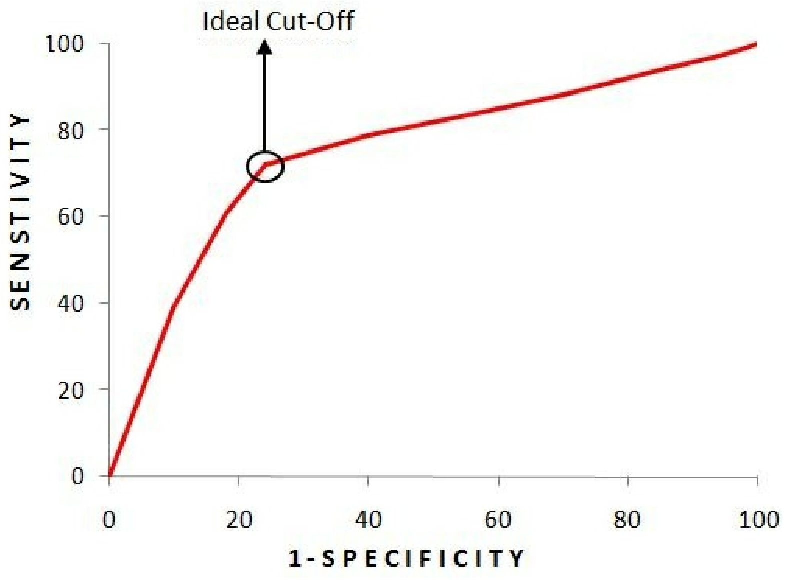
Specificity. Now all these measures depend on the values that we have as True Positives, True Negatives, False Positive and False Negatives which can be manipulated by altering the threshold value. For example, if we raise the threshold value to 90% then it will become tough for an observation to be categorised as 'Non-Defaulter' and this will affect the various measures.

Thus, we can say that the True Positive and False Positive Rates are simply a function of the threshold and for different values of threshold we will have different values of such measures. This is where we take help of Receiver Operating Characteristic Curve or ROC Curve which is originally based on a military communication device which was used to communicate through radio and morse code. With the help of ROC Curve, we consider different threshold values and get to know the different values of True Positive Rate (Sensitivity) and False Positive Rate (1-Specificity).

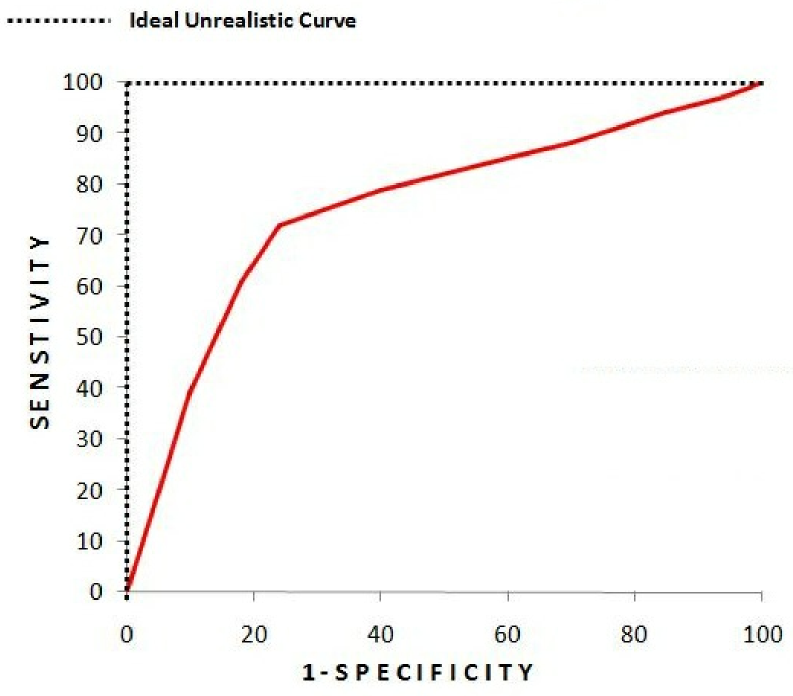
Sensitivity v/s 1-Specificity

On one hand, we have Sensitivity which provides us with the percentage of positives that were classified correctly by the model and naturally we want this number to be high. On the other hand, we have Specificity which indicates how good the model is at avoiding the misclassification of the negatives i.e. out of the predicted negatives, how many of them are actually negatives and naturally we want this number to be as high as possible. Now if these values are high then it means that the diagonal values in the confusion matrix are high indicating that the number of False Positive and Negative observations are low. Now we want to see for which value of threshold these two numbers are the highest, however, as we want both Sensitivity and Specificity to be as high as possible, it makes it difficult to plot their values on a graph for different values of thresholds. Thus, rather than looking for very high Specificity we look for very low False Positive Rate which indicates the percentage of negatives that were classified as positive by the model (basically an opposite of Specificty i.e. 1-Specificity).

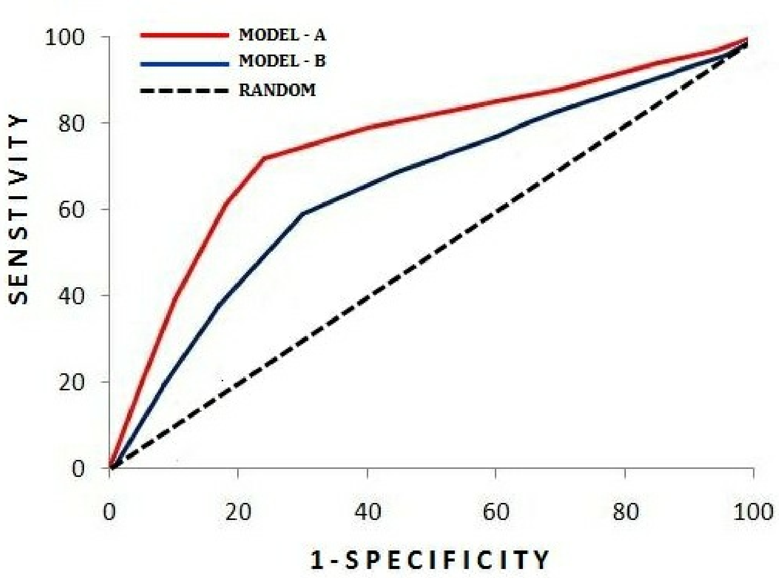
Now for different levels of Threshold when we plot the values of Sensitivity and 1-Specificity we get the following graph.



Here for the Logistic Regression Model, we get different values of specificity and sensitivity for different values of the thresholds. The ideal scenario will be where the value of threshold that provides us with 100 sensitivity and 0 1-specificity, however, as it is the most ideal case which is generally not possible, we pick the next best thing which is the ideal cutoff where the difference between the two is highest i.e. we pick that value of threshold where we can have the maximum sensitivity with minimum 1-specificity. (The point closest from the upper left corner, the probability corresponding to it gives the best cutoff)

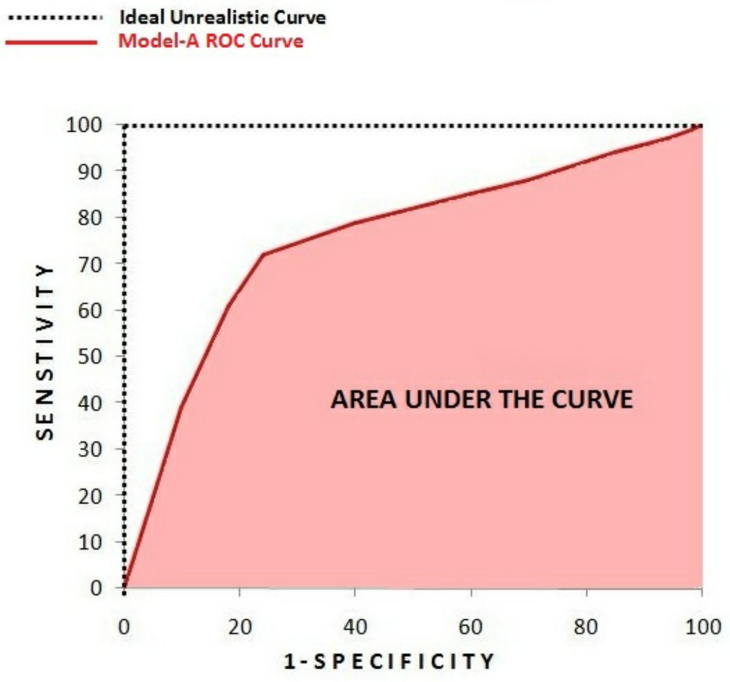


ROC Curve can also be used to compare two models as stand-alone values of different measures cannot make much sense to compare two models. Below we see for different values of the threshold i.e. the values of Sensitivity and 1-Specificity for Model-A (Logistic Regression) and Model-B (Decision Trees). Here no matter which threshold we consider for Model-A, it will provide better results than results provided by the ideal cut-off of Model-B. The dotted black line represents a ROC curve of a completely random classifier. For example, if we have a dataset which has an equal number of class label '0' and '1' then a model which predicts all of them as 0 will have this dotted line as shown in the ROC curve.



AUC Curve

It is a single number metric which is an extension of the ROC Curve. Here we measure the area under the curve. For the perfect unrealistic curve, the area under the curve will be 100%. While for the completely random model the area under the curve will be 50%. A model which performs worse than a random model will have an area below 50%. Here Model-A will be more accurate than Model-B as the area under the curve is more for Model-A than it is for Model-B. Thus, AUC Curve provides us with a single value metric which quickly helps us to understand the performance of the classification model.

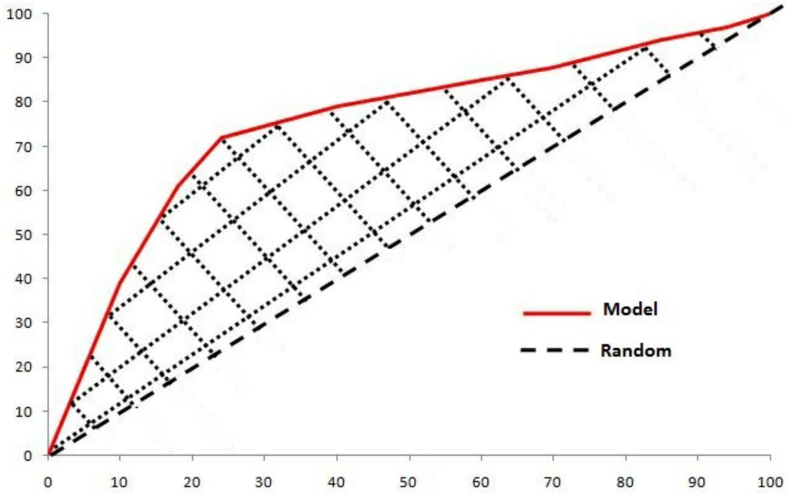


If we compare a base rate model with a model which is more accurate then the difference between their AUC score will be starker than the difference between their Sensitivity.

Thus, AUC score provides us with a better idea of how the model is working as with the ROC curve sometimes the difference between the two models may not be very visible, with the ROC curves of different models intersecting each other with models performing better in some regions than other. AUC score provides us with good overall evaluation.

Gini Coefficient

Gini Coefficient tells us about the differentiating power of the model. In a ROC Curve, it is the area between the random model and the created model (ROC Curve).



The area between the random and model line tells us the additional lift that the model is able to get. The formula for Gini coefficient is 2 × AUC-1

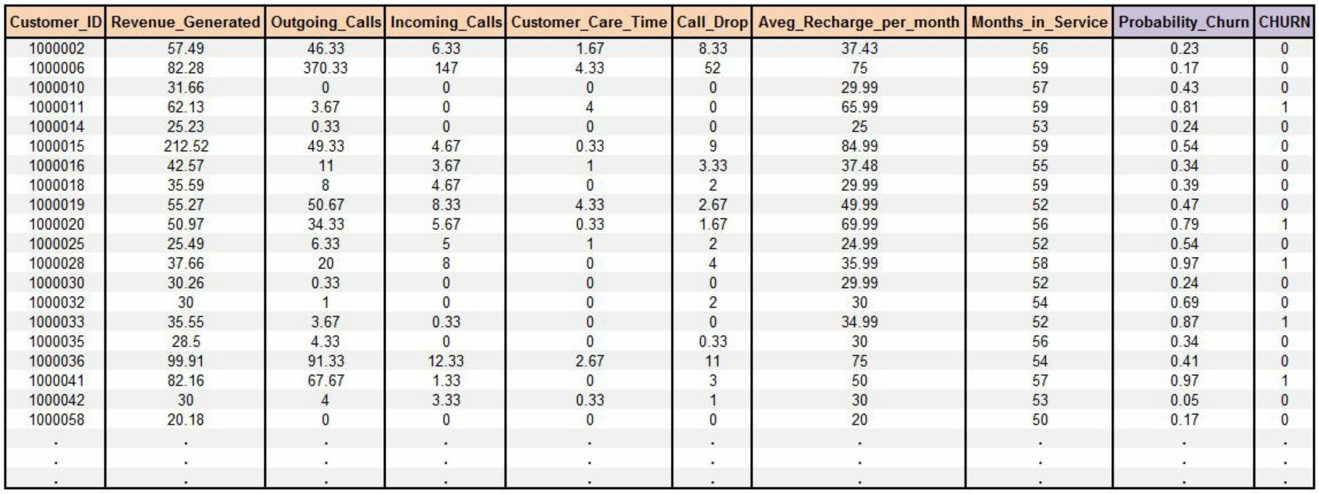
Gain and Lift Charts

Unlike the Confusion Matrix, Gain and Lift charts help us in visualising the performance of our model in comparison to the base model/no model. Gain and Lift charts can help us in understanding how our model is performing on different sections of the data.
To understand how Gain and Lift charts work, let us first take an example of a telecom company who wants to engage with its customers to reduce its attrition/churn rate or in simple words, to stop the customers from discontinuing with the telecom company.

A prediction from a Base Model or by having no model provides us with a 10% iteration rate i.e on average by the end of every year 10% of the company's current customer discontinue the network. Thus, if the telecom company has 100,000 customers then 10,000 of them will discontinue the subscription. However, the company has limited resources and cannot contact all of its customer bases and therefore requires some sort of prioritization of the customers so that the company can divert its resources to those customers who are going to churn.

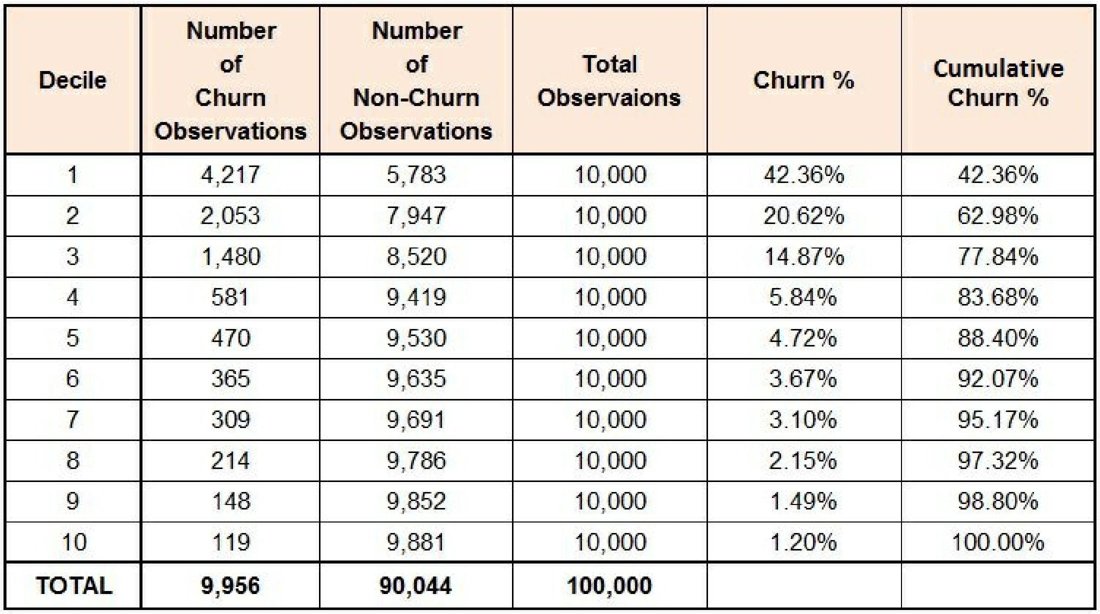
We have the data of customers of past 1 year and we build a logistic regression model to find the customers who eventually unsubscribed. Now in addition to building a confusion matrix and ROC/AUC curve, we can also create a Gain and Lift Chart.

For every observation (details of a customer), the logistic regression model provides us with the probability of that observation being categorised as 1 "Churn / Unsubscribed". From the analysis of the ROC curve, we decide to go with a cut-off value of 0.7 which means that any observation that has the churn probability of more than 70% will be declared as a customer who unsubscribed.



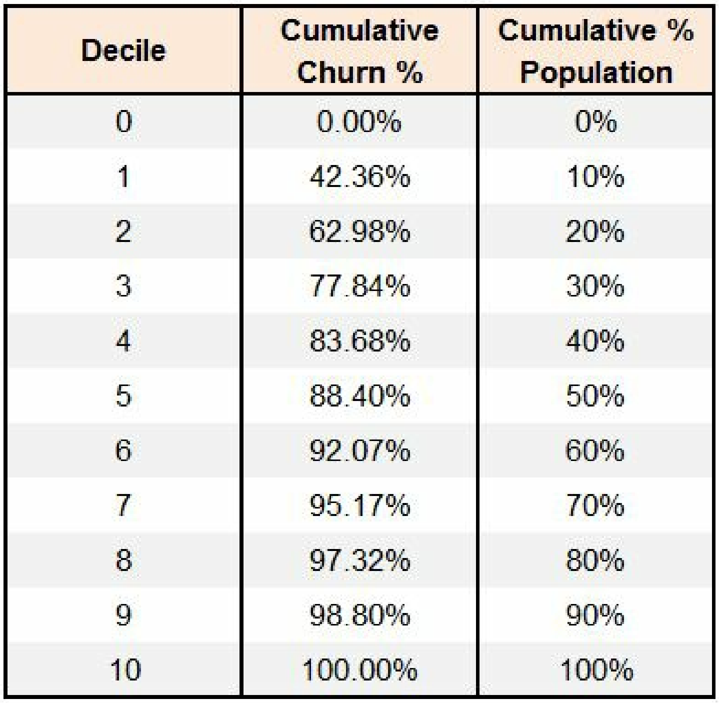
Customer_ID	Revenue_Generated	Outgoing_Calls	Incoming_Calls	Customer_Care_Time	Call_Drop	Aveg_Recharge_per_month	Months_in_Service	Probability_Churn	CHURN
1000002	57.49	46.33	6.33	1.67	8.33	37.43	56	0.23	0
1000006	82.28	370.33	147	4.33	52	75	59	0.17	0
1000010	31.66	0	0	0	0	29.99	57	0.43	0
1000011	62.13	3.67	0	4	0	65.99	59	0.81	1
1000014	25.23	0.33	0	0	0	25	53	0.24	0
1000015	212.52	49.33	4.67	0.33	9	84.99	59	0.54	0
1000016	42.57	11	3.67	1	3.33	37.48	55	0.34	0
1000018	35.59	8	4.67	0	2	29.99	59	0.39	0
1000019	55.27	50.67	8.33	4.33	2.67	49.99	52	0.47	0
1000020	50.97	34.33	5.67	0.33	1.67	69.99	56	0.79	1
1000025	25.49	6.33	5	1	2	24.99	52	0.54	0
1000028	37.66	20	8	0	4	35.99	58	0.97	1
1000030	30.26	0.33	0	0	0	29.99	52	0.24	0
1000032	30	1	0	0	2	30	54	0.69	0
1000033	35.55	3.67	0.33	0	0	34.99	52	0.87	1
1000035	28.5	4.33	0	0	0.33	30	56	0.34	0
1000036	99.91	91.33	12.33	2.67	11	75	54	0.41	0
1000041	82.16	67.67	1.33	0	3	50	57	0.97	1
1000042	30	4	3.33	0.33	1	30	53	0.05	0
1000058	20.18	0	0	0	0	20	50	0.17	0
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.

As Gain and Lift charts are based on the ordering of these probabilities, we sort the probabilities in decreasing order. Then we perform deciling or grouping of the data. We created 10 deciles with each decile having 10% of the data. We then calculate the number of churn observations captured in each decile and come up with the following table.

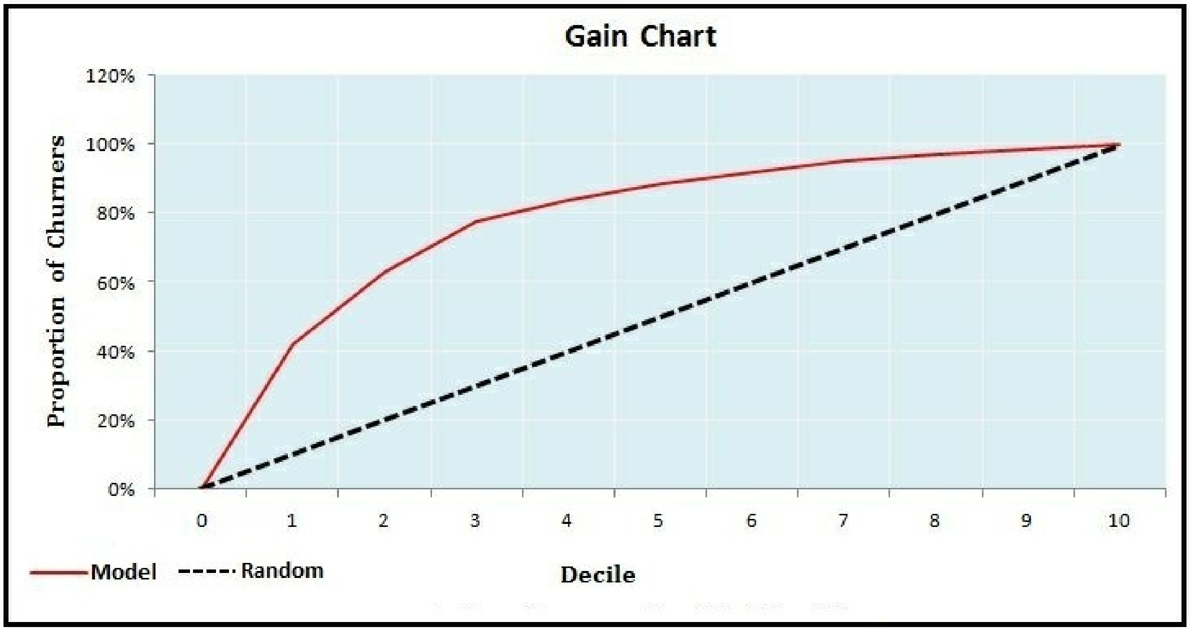


Decile	Number of Churn Observations	Number of Non-Churn Observations	Total Observaions	Churn %	Cumulative Churn %
1	4,217	5,783	10,000	42.36%	42.36%
2	2,053	7,947	10,000	20.62%	62.98%
3	1,480	8,520	10,000	14.87%	77.84%
4	581	9,419	10,000	5.84%	83.68%
5	470	9,530	10,000	4.72%	88.40%
6	365	9,635	10,000	3.67%	92.07%
7	309	9,691	10,000	3.10%	95.17%
8	214	9,786	10,000	2.15%	97.32%
9	148	9,852	10,000	1.49%	98.80%
10	119	9,881	10,000	1.20%	100.00%
TOTAL	**9,956**	**90,044**	**100,000**		

As per the table, we are able to capture almost 84% of the data in first four deciles. Before drawing any more inferences, we should create a gain chart which is line graph between cumulative churn % and cumulative population %.

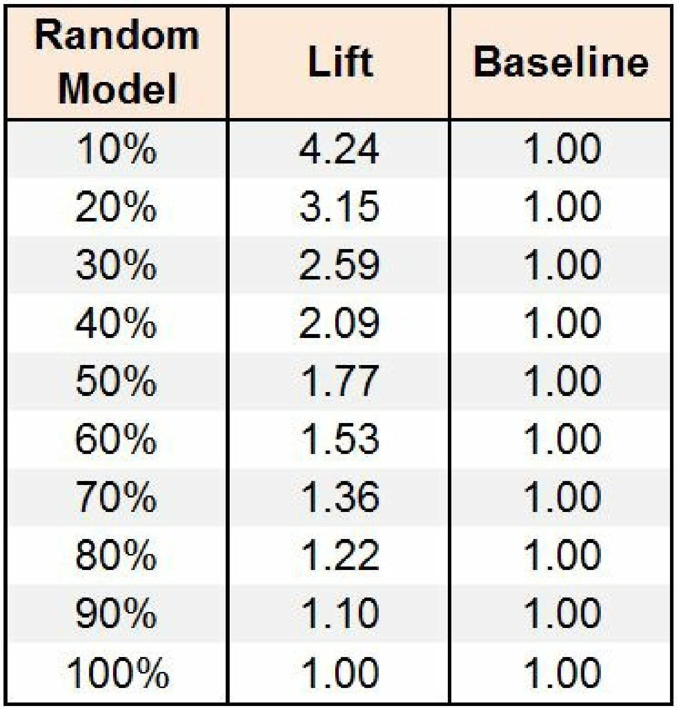


Decile	Cumulative Churn %	Cumulative % Population
0	0.00%	0%
1	42.36%	10%
2	62.98%	20%
3	77.84%	30%
4	83.68%	40%
5	88.40%	50%
6	92.07%	60%
7	95.17%	70%
8	97.32%	80%
9	98.80%	90%
10	100.00%	100%

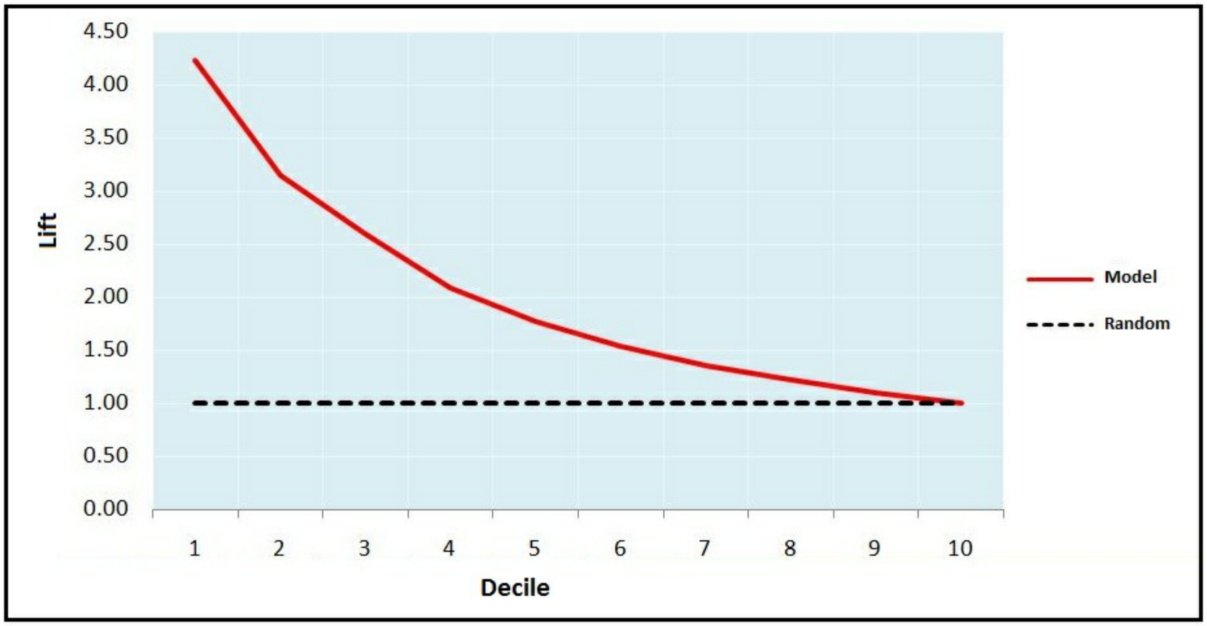


If we use a random model then each decile will have an equal number of churners and the rate of capturing churners will increase 10% perpetually for each decile. Thus, for the first decile, among all the churners only 10% will be captured and similarly for the second decile we will capture another 10% making the cumulative to 20% and so on. Therefore, the rate of capture will be equal to the percentage of observations in each decile. This will make prioritizing of customers very difficult. However, when we use a model, we are able to capture 42% of 'churners' in the first decile. The first decile only has 10% of the customers and from these 10% customers, the model is able to capture 42% of the churners. A random model here would have been able to capture only 10% of the churners. This 32% is the 'gain' or additional information that the model has provided. Thus, the rate of capturing 'churners' is way higher when we use a model.

In the first decile, a random model would have captured 10% of the 'churners' while our model captures 42%. This means a lift of 4.24 times, therefore, by contacting only 10% of our customers, we will be able to reach 4 times the 'churn customers' when compared to a random model/no model.

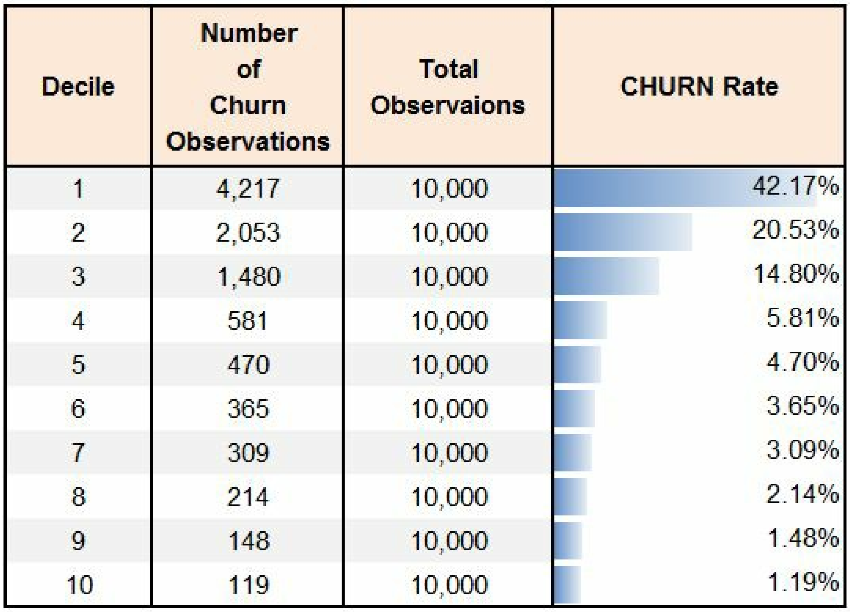


Random Model	Lift	Baseline
10%	4.24	1.00
20%	3.15	1.00
30%	2.59	1.00
40%	2.09	1.00
50%	1.77	1.00
60%	1.53	1.00
70%	1.36	1.00
80%	1.22	1.00
90%	1.10	1.00
100%	1.00	1.00



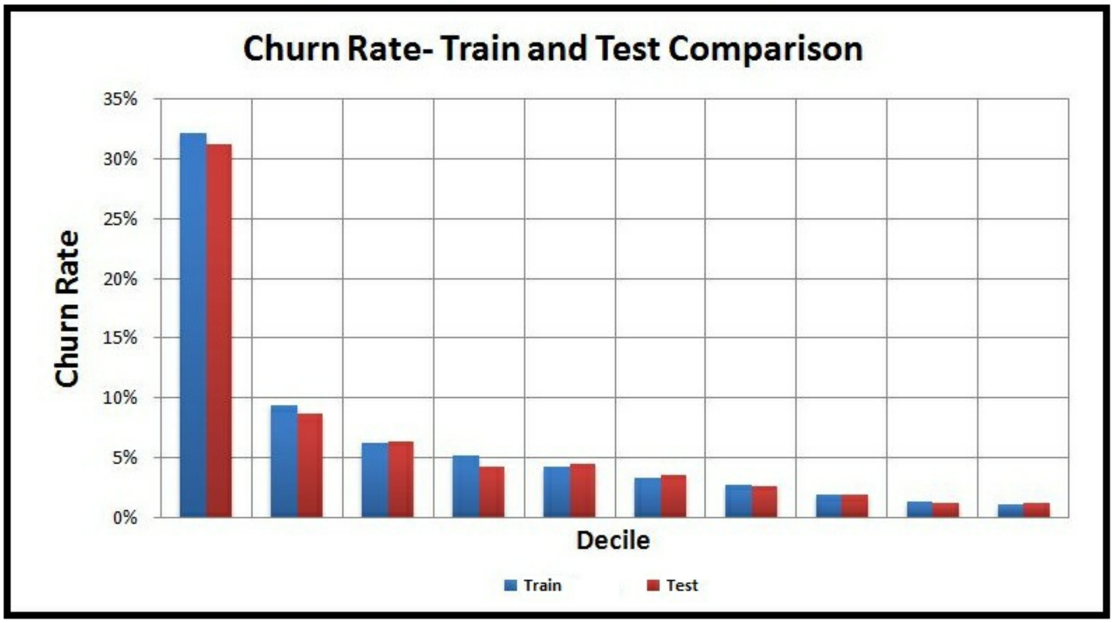
Lift charts help us in targeting customers and plan campaign by narrowing down our target population. However, our main aim here is to evaluate our model and to do that we have to look for certain things which tell us how well the model is performing such as, the Lift generated in the first two deciles for a good model should be more than 2.0. In the gain chart, the area that falls between the random model line and the gain line can be used as a measure to evaluate a model.

Also, the Numbers of Churn observations and Churn Rate should perpetually fall with each decile (should follow rank ordering for at least 5 or 6 deciles). Here churn rate for each decile will be Number of Churn observation divided by the number of observations in that decile.



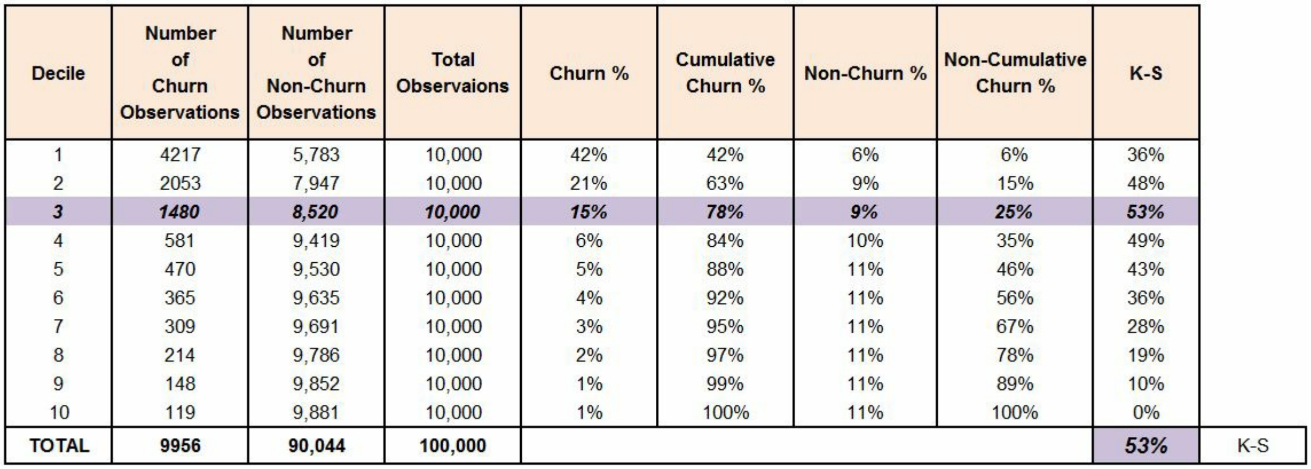
Decile	Number of Churn Observations	Total Observaions	CHURN Rate
1	4,217	10,000	42.17%
2	2,053	10,000	20.53%
3	1,480	10,000	14.80%
4	581	10,000	5.81%
5	470	10,000	4.70%
6	365	10,000	3.65%
7	309	10,000	3.09%
8	214	10,000	2.14%
9	148	10,000	1.48%
10	119	10,000	1.19%

We can also make sure that the model created on Train and Test dataset have a similar percentage of churners (churn rate) in each decile and it’s better if the lines of both the models coincide with each other in the Gain and Lift chart.



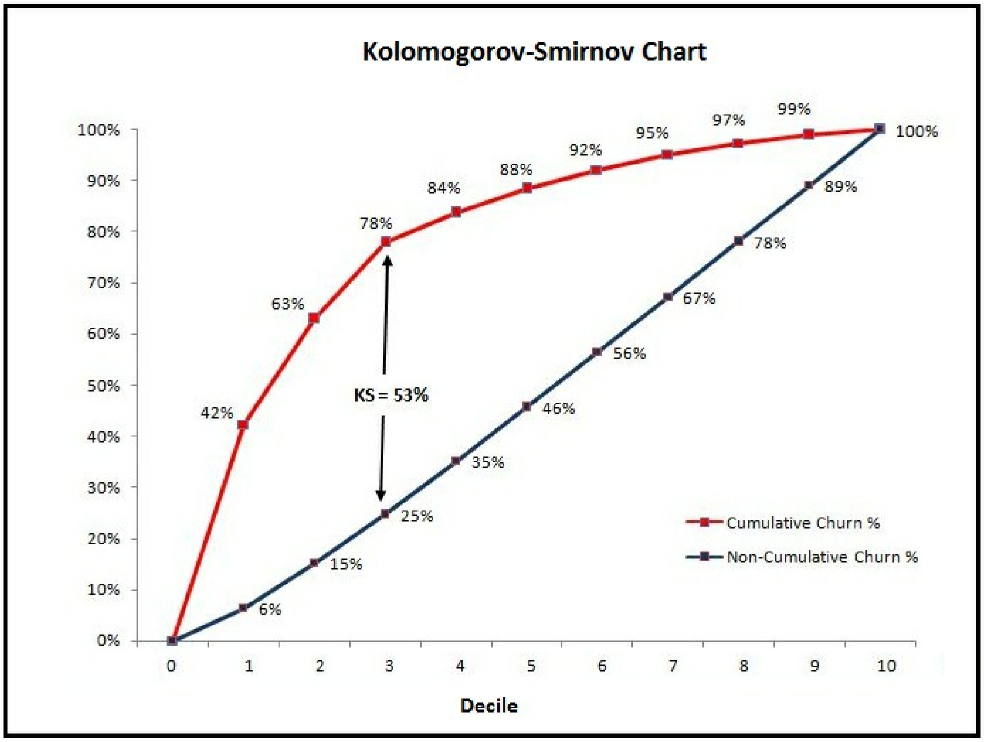
KS Chart (Kolmogorov-Smirnov)

If we continue the above example, then a Kolmogorov-Smirnov Chart will help us to understand how good our model is in differentiating the customers who are going to discontinue (Churners) from the customers that are going to stay (Non-Churners). KS statistic is the difference between the cumulative target and the cumulative non-target. In our example, as our target is churn then KS statistic will be the difference between the Cumulative Churn and Cumulative Non-Churn observations. Higher K-S value means that the model is good at separating the two classes. A K-S statistic=100 will mean that the model is able to create two mutually exclusive groups with each group having a separate class label of observations. K-S Statistic=0 indicates a very poor model which fails to successfully distinguish between the classes.



Decile	Number of Churn Observations	Number of Non-Churn Observations	Total Observaions	Churn %	Cumulative Churn %	Non-Churn %	Non-Cumulative Churn %	K-S	
1	4217	5,783	10,000	42%	42%	6%	6%	36%	
2	2053	7,947	10,000	21%	63%	9%	15%	48%	
3	***1480***	***8,520***	***10,000***	***15%***	***78%***	***9%***	***25%***	***53%***	
4	581	9,419	10,000	6%	84%	10%	35%	49%	
5	470	9,530	10,000	5%	88%	11%	46%	43%	
6	365	9,635	10,000	4%	92%	11%	56%	36%	
7	309	9,691	10,000	3%	95%	11%	67%	28%	
8	214	9,786	10,000	2%	97%	11%	78%	19%	
9	148	9,852	10,000	1%	99%	11%	89%	10%	
10	119	9,881	10,000	1%	100%	11%	100%	0%	
TOTAL	**9956**	**90,044**	**100,000**					***53%***	K-S

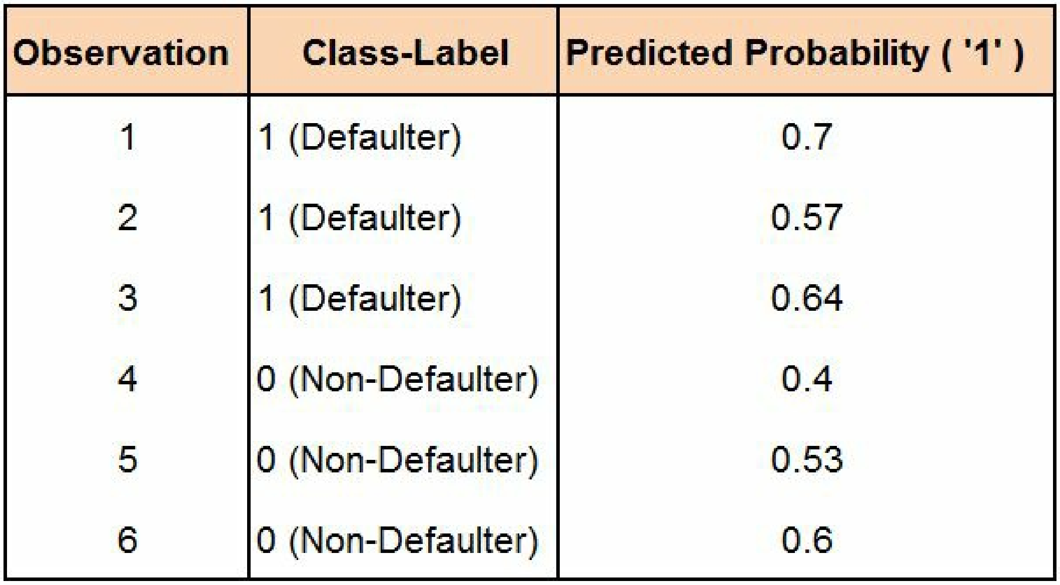
We can see that the maximum separation that we receive is in the third decile. For a good model, the maximum K-S statistic should fall in the top three or four deciles as we expect the maximum differentiation between the churners and non-churners to happen in the initial deciles only.



In our example, the KS static comes out to be 53%.

Concordance

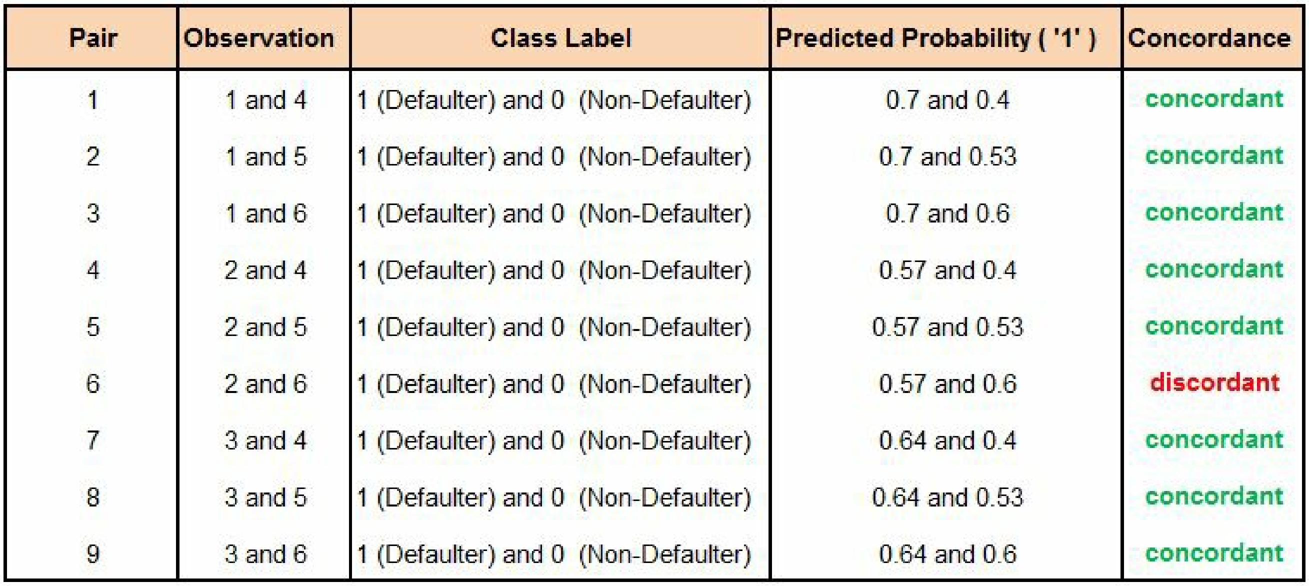
It is one of the single most important metrics when evaluating a classification model's performance. To understand concordance, we can use an example where we have six observations with three having class label 1 ('Defaulters') and class label '0' (Non-Defaulters). We then build a logistic regression model that provides us with the probability of observations being of class label '1'.



Observation	Class-Label	Predicted Probability ('1')
1	1 (Defaulter)	0.7
2	1 (Defaulter)	0.57
3	1 (Defaulter)	0.64
4	0 (Non-Defaulter)	0.4
5	0 (Non-Defaulter)	0.53
6	0 (Non-Defaulter)	0.6

Our model will be concordant when among all the pairs formed from '0' and '1' observations, the percentage of pairs where the probability assigned to observations with the class label '1' is greater than the observations with the class label '0'.

To put all this in the context, we first make all possible combinations of class labels '0' and '1' along with the predicted probabilities for them. In our example, we will form 9 such combinations/pairs.



Pair	Observation	Class Label	Predicted Probability ('1')	Concordance
1	1 and 4	1 (Defaulter) and 0 (Non-Defaulter)	0.7 and 0.4	concordant
2	1 and 5	1 (Defaulter) and 0 (Non-Defaulter)	0.7 and 0.53	concordant
3	1 and 6	1 (Defaulter) and 0 (Non-Defaulter)	0.7 and 0.6	concordant
4	2 and 4	1 (Defaulter) and 0 (Non-Defaulter)	0.57 and 0.4	concordant
5	2 and 5	1 (Defaulter) and 0 (Non-Defaulter)	0.57 and 0.53	concordant
6	2 and 6	1 (Defaulter) and 0 (Non-Defaulter)	0.57 and 0.6	discordant
7	3 and 4	1 (Defaulter) and 0 (Non-Defaulter)	0.64 and 0.4	concordant
8	3 and 5	1 (Defaulter) and 0 (Non-Defaulter)	0.64 and 0.53	concordant
9	3 and 6	1 (Defaulter) and 0 (Non-Defaulter)	0.64 and 0.6	concordant

For the first pair, the probability of an actual defaulter to default is higher than the probability of an actual non-defaulters to default making us correctly categorising this observation. Similarly we form 9 such pairs, however, in the Pair 6, we get a discordant pair. Thus, out of the 9, 8 pairs are concordant while 1 is discordant. Concordance is a ratio of such concordant and discordant pairs. In this example, the concordance comes out to be 88.9% ((8 ÷ 9) × 100).

If the probabilities are same (for example: 0.55 and 0.55), then in such a scenario the pair will be considered a tie and will neither be concordant nor discordant.

Thus, the model's segregation power matters in this metric as concordance helps us to understand how well the model segregates the negatives from the positives. Minimum accepted concordance for a model to be considered worthy is 60%.

All the metrics discussed so far played a crucial role in order to evaluate the performance of a model. We use these metrics during the training phase and after evaluating a model's performance on the training dataset, we apply the model on the test set and again conduct a series of tests using the above-mentioned measures to evaluate the model. However, we need to validate a model in order to properly understand how well it performs with unseen or test data. To evaluate model's performance with unseen data, we can perform various model validation methods explored in the section 'Model Validation'.