**1. In the sense of machine learning, what is a model? What is the best way to train a model?**

Ans: A machine learning model is a file that has been trained to recognize certain types of patterns. You train a model over a set of data, providing it an algorithm that it can use to reason over and learn from those data.

Once you have trained the model, you can use it to reason over data that it hasn't seen before, and make predictions about those data

Best way to train Model:

1. Begin with existing data
2. Analyse data to identify patterns
3. Make predictions

**2. In the sense of machine learning, explain the "No Free Lunch" theorem.**

Ans: The “no free lunch” (NFL) theorem for supervised machine learning is a theorem that essentially implies that no single machine learning algorithm is universally the best-performing algorithm for all problems

Mathematically, the computing cost of finding a solution is the same for any solution technique when averaged across all problems in the class. As a result, no solution provides a shortcut.

Machine learning models adhere to the Garbage in, Garbage out (GIGO) principle (i.e. Predictions rely on the data quality on which our model is trained). And a lot of study went into these theorems, and others may claim that this theorem does not apply in all instances. It is preferable that we concentrate on the aspects that will help us better comprehend the data and construct the best performing models.

**3. Describe the K-fold cross-validation mechanism in detail.**

Ans: Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation.

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model.

The general procedure is as follows:

1. Shuffle the dataset randomly.
2. Split the dataset into k groups
3. For each unique group:
   1. Take the group as a hold out or test data set
   2. Take the remaining groups as a training data set
   3. Fit a model on the training set and evaluate it on the test set
   4. Retain the evaluation score and discard the model
4. Summarize the skill of the model using the sample of model evaluation scores

Importantly, each observation in the data sample is assigned to an individual group and stays in that group for the duration of the procedure. This means that each sample is given the opportunity to be used in the hold out set 1 time and used to train the model k-1 times.

It is a popular method because it is simple to understand and because it generally results in a less biased or less optimistic estimate of the model skill than other methods, such as a simple train/test split.

**4. Describe the bootstrap sampling method. What is the aim of it?**

Ans: Bootstrapping is a method of inferring results for a population from results found on a collection of smaller random samples of that population, using replacement during the sampling process.

The main aim is that bootstrapping saves a lot of time during the phase of conducting research when it is too difficult, time-consuming or costly to survey the entire population being looked at.

**5. What is the significance of calculating the Kappa value for a classification model? Demonstrate how to measure the Kappa value of a classification model using a sample collection of results.**

Ans: The kappa score is an interesting metric. Its origins are in the field of psychology: it is used for measuring the agreement between two human evaluators or raters (e.g., psychologists) when rating subjects (patients). It was later “appropriated” by the machine-learning community to measure classification performance. Sometimes in machine learning we are faced with a multi-class classification problem. In those cases, measures such as the accuracy, or precision/recall do not provide the complete picture of the performance of our classifier.

In some other cases we might face a problem with imbalanced classes. E.g. we have two classes, say A and B, and A shows up on 5% of the time. Accuracy can be misleading, so we go for measures such as precision and recall. There are ways to combine the two, such as the F-measure, but the F-measure does not have a very good intuitive explanation, other than it being the harmonic mean of precision and recall.

cohen’s kappa statistic is a very good measure that can handle very well both multi-class and imbalanced class problems.

Cohen’s kappa is defined as:

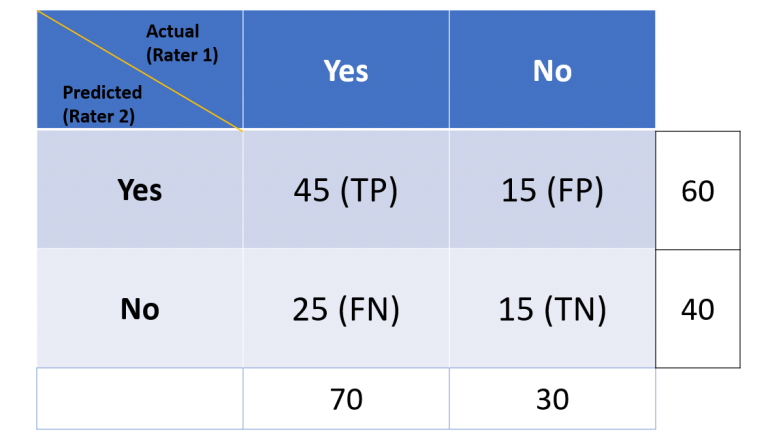


where po is the observed agreement, and pe is the expected agreement. It basically tells you how much better your classifier is performing over the performance of a classifier that simply guesses at random according to the frequency of each class.Cohen’s kappa is always less than or equal to 1. Values of 0 or less, indicate that the classifier is useless. There is no standardized way to interpret its values.

How to calculate Cohen’s Kappa Score?

Cohen’s Kappa can be calculated using either raw data or confusion matrix values. When Cohen’s Kappa is calculated using raw data, each row in the data represents a single observation, and each column represents a rater’s classification of that observation. Cohen’s Kappa can also be calculated using a confusion matrix, which contains the counts of true positives, false positives, true negatives, and false negatives for each class. We will look into the details of how the Kappa score can be calculated using a confusion matrix.

confusion matrix representing a binary classification model where there are two classes / labels:



* **True Positives (TP)** – These are the correctly predicted positive values. In this case, both the raters (observer in real-world and classification model) are in agreement.
* **False Positives (FP)** – These are the incorrectly predicted positive values. In other words, these are instances which are predicted as positive although they are truly negative. Observer and classification are in disagreement about the true negatives. These are instances which are called out negative by the observer but the positive (falsely) by the classification model.
* **True Negatives (TN)** – These are the correctly predicted negative values. The observer and the classification are in perfect agreement about the negatives.
* **False Negatives (FN)** – These are incorrectly predicted negative values. The observer and the classification models are in disagreement about the true positives. These are instances which are called out positive by the observer but the negative (falsely) by the classification model.

In the above confusion matrix, the **actual** represents **rater 1**. Rater 1 is an observer of real world events and record what actually happened. The **predicted** represents **rater 2.** The rater 2 represents the classification model which makes the predictions. Cohen Kappa score will be used to assess the model performance as a function of probability that the rater 1 and rater 2 are in perfect agreement (TP + TN), also denoted as Po (observed probability), and, the probability (expected) both the raters are in agreement by chance or randomly.

Let the total observation (TP + FP + FN + TN) is N. Or, **N = TP + FP + FN + TN**

**first step is to calculate the probability that both the raters are in perfect agreement:**

**Observed Agreement, Po = (TP + TN) / N**

**In our example, this would be:**

**Po = (45+15)/100=0.6**

Next, we need to calculate the expected probability that both the raters are in agreement by chance. This is calculated by multiplying the expected probability that both the raters are in agreement that the classes are positive, and the classes are negative.

**Pe = [{Pe(rater 1 says Yes) / N}\* {Pe(rater 2 says Yes) / N} + [{Pe(rater 1 says no) / N} \* {Pe(rater 2 says no) / N}]**

**So in our case this would be calculated as:**

**Pe = 0.7 x 0.6 + 0.4 x 0.3 = 0.42 + 0.12 = 0.54**

**Now that we have both the observed and expected agreement, we can calculate Cohen’s Kappa:**

**Kappa score = (Po – Pe) / (1 – Pe)**

**In our example, this would be:**

**K = (0.6 – 0.54)/(1 – 0.54)= 0.06 / 0.46 = 0.1304 or a little over 13%**

Kappa can range from 0 to 1. A value of 0 means that there is no agreement between the raters (real-world observer vs classification model), and a value of 1 means that there is perfect agreement between the raters. In most cases, anything over 0.7 is considered to be very good agreement.

**6. Describe the model ensemble method. In machine learning, what part does it play?**

Ans: Ensemble modeling is the process of running two or more related but different analytical models and then synthesizing the results into a single score or spread in order to improve the accuracy of predictive analytics and data mining applications

Ensemble methods is a machine learning technique that combines several base models in order to produce one optimal predictive model

**7. What is a descriptive model's main purpose? Give examples of real-world problems that descriptive models were used to solve.**

Ans: The descriptive model definition states that modeling is a way to connect mathematics with real-world situations in order to solve problems in finance, industry, public policy, and the biological and physical sciences.

A descriptive model describes a system or other entity and its relationship to its environment. It is generally used to help specify and/or understand what the system is, what it does, and how it does.

Descriptive models help the user to identify the different factors that contribute to a situation, event, or reaction. Moreover, descriptive models enable individuals to generate ideas for problem-solving with applications in the real world. This process thus enables the user to better understand potentially complex scientific, mathematical, or social phenomena.

Example: students need Rs .1500 to cover the cost of travel and lodging for a trip. In order to raise money students are selling Pizza-making kits. Each kit raises Rs. 20 for the trip. How many pizza kits will the students need to sell in order to raise enough money to send the entire class on a trip.

So, Total cost for 10 students = 10\*1500 = Rs. 15000

To determine no. of pizza kit to sell: 15000/20 = 750 kits

Using this calculation students set a goal to sell 750 pizza kits to raise the funds for their trip.

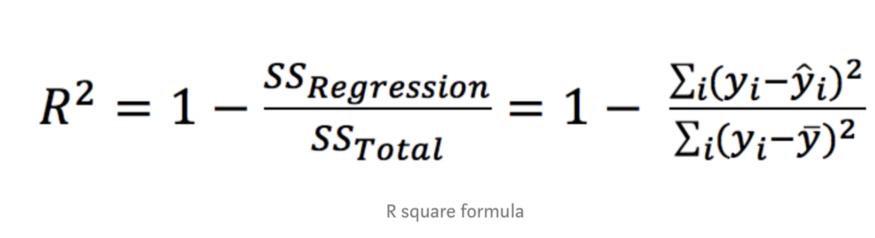
The steps used by the students to calculate the number of pizza kits to sell for their class trip represent an example of a descriptive model.

**8. Describe how to evaluate a linear regression model.**

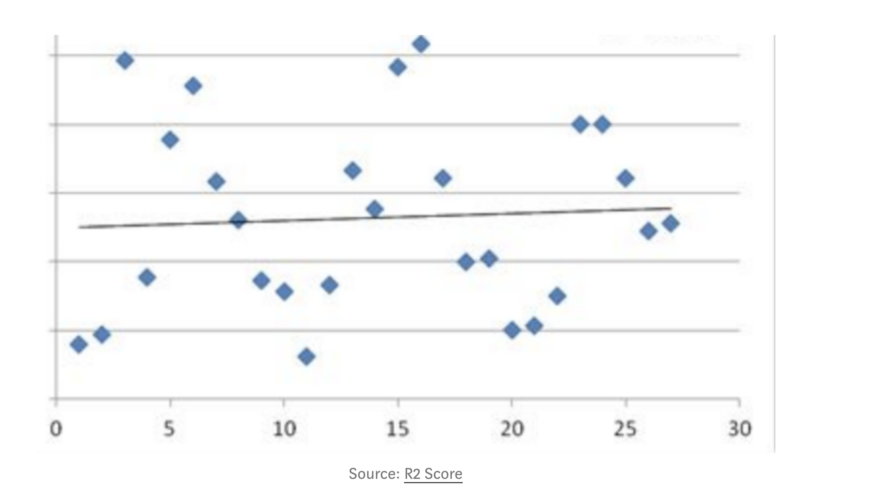
Ans: To evaluate models. we use metrics like

1. R Square/Adjusted R Square
2. Mean Square Error(MSE)/Root Mean Square Error(RMSE)
3. Mean Absolute Error(MAE)
4. illustrate Residual of model as a normal distribution ( bell shape)

1) R Square/Adjusted R Square:

This is a first measure of the regression model especially we, everybody, do during evaluation because it is easy to interpret scores between 0 to 1. If we see a good score like close to 1, then we assume that model is a good fit. Of course , R Square is a good measure to determine how well the model fits the dependent variables. However, it does not take into consideration the overfitting problem. If your regression model has many independent variables, because the model is too complicated, it may fit very well to the training data but performs badly for testing data.So I recommend that we have to see all perspectives for better evaluation . Let's talk about what R 2 actually means . R² is calculated by the sum of squares of prediction error divided by the total sum of squares which replace the calculated prediction with mean. R Square value is between 0 to 1 and bigger value indicates a better fit between prediction and actual value.

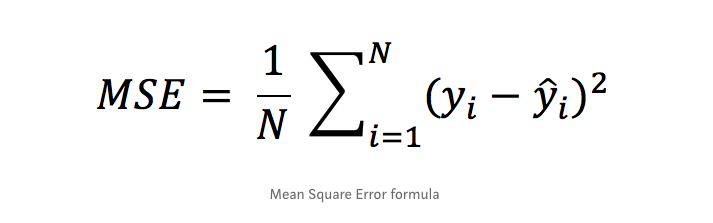
Sometimes , R² is more helpful to measure error on a model than Mean Square Error(MSE) and Mean Absolute Error(MAE). For instance , We can say R² is a perfect measure to give you a model like in the figure below.



Here , I also want to focus R² Adjust measure too. Sometimes, We see R² and R² Adjust same score. But When we do fine tuning to model to get better accuracy then R² Adjust helps us to better understand. It happens when we add more independent features or penalize more features due to over fitting . Then we can see different scores between these measures.

2) Mean Square Error(MSE)/Root Mean Square Error(RMSE):

while R² is a relative measure of how well the model fits dependent variables, whereas Mean Square Error is an absolute measure of the fit of the model. MSE is calculated by the sum of squares of prediction error. Where prediction error is minus between true values and prediction values, and then it is made square because we avoid negative error scores. Its result gives us how much deviation from the actual number. Its number might be larger, which may be uncommon . you might be questioning how the error score is too big .



For example

print(mean\_squared\_error(Y\_test, Y\_pred))

print(math.sqrt(mean\_squared\_error(Y\_test, Y\_pred)))

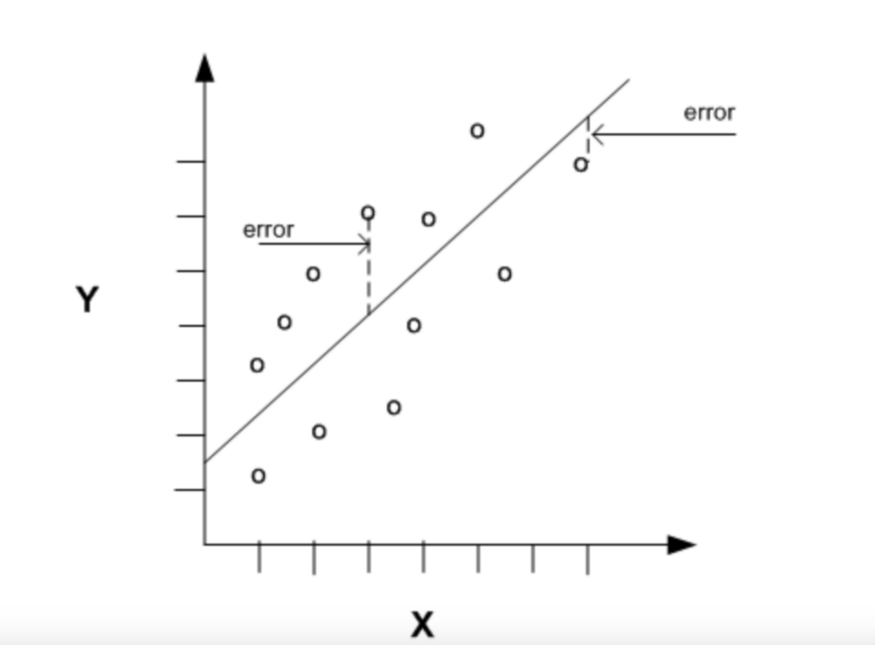
# MSE: 109.86374118394116

# RMSE: 10.48159058463653

For example, we can see MSE is too big . So for this solution, we can use the Root Mean Square Error metric which gives better interpretation about the model. Again , One more question comes into our mind that how the number represents a good or bad fit model. According to theory, if MSE or RMSE is 0.0 , then it has no error. But in real life projects , we never get this 0.0 error. We always get some error score , now how to evaluate the number. First of all , we have to calculate the mean value of dependent variables . Then we can compare the mean value of the dependent variable and RMSE error score . After that , we can see some percentage of deviation from real value. For example: we take dependent values from the below example where dependent values are nearly close to mean value 499.31 and the RMSE model is 10.48 . So it is a 2.09 % difference between true values and prediction values.

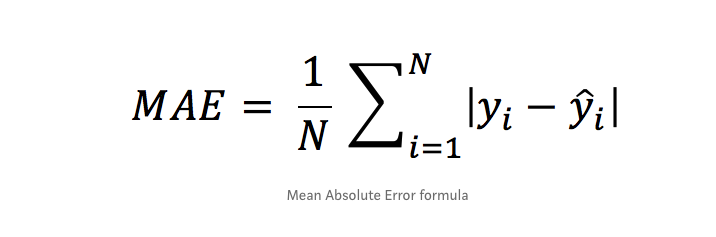
print(mean\_squared\_error(Y\_test, Y\_pred))

y = df['Yearly Amount Spent']



3) Mean Absolute Error(MAE):

This is almost the same as the Mean Square Error metric but only MAE takes absolute error value instead of square of predicted error for avoiding negative score . However, here , we don’t need to calculate the Root of MAE score . We can directly interpret the score with real values.



4) Explore Residual :

At last for evaluation , We can also explore residuals, which comes from true values and predicted values, by scatter plot or distplot of seaborn library or matplotlib. If we get a linear shape on a scatter plot or bell shape in a distplot , then we can pretty much say that model fits perfectly, and can predict very close to real values. For Example :

# explore residual

residual = y\_test - y\_pred

sns.distplot(residual)

**9. Distinguish :**

1. Descriptive vs. predictive models

Ans: A descriptive model will exploit the past data that are stored in databases and provide you with the accurate report. In a Predictive model, it identifies patterns found in past and transactional data to find risks and future outcomes.

2. Underfitting vs. overfitting the model

Underfitting: A statistical model or a machine learning algorithm is said to have underfitting when it cannot capture the underlying trend of the data, i.e., it only performs well on training data but performs poorly on testing data. (It’s just like trying to fit undersized pants!) Underfitting destroys the accuracy of our machine learning model. Its occurrence simply means that our model or the algorithm does not fit the data well enough. It usually happens when we have fewer data to build an accurate model and also when we try to build a linear model with fewer non-linear data. In such cases, the rules of the machine learning model are too easy and flexible to be applied to such minimal data and therefore the model will probably make a lot of wrong predictions. Underfitting can be avoided by using more data and also reducing the features by feature selection.

In a nutshell, Underfitting refers to a model that can neither perform well on the training data nor generalize to new data.

Reasons for Underfitting:

* High bias and low variance
* The size of the training dataset used is not enough.
* The model is too simple.
* Training data is not cleaned and also contains noise in it.

Techniques to reduce underfitting:

* Increase model complexity
* Increase the number of features, performing feature engineering
* Remove noise from the data.
* Increase the number of epochs or increase the duration of training to get better results.

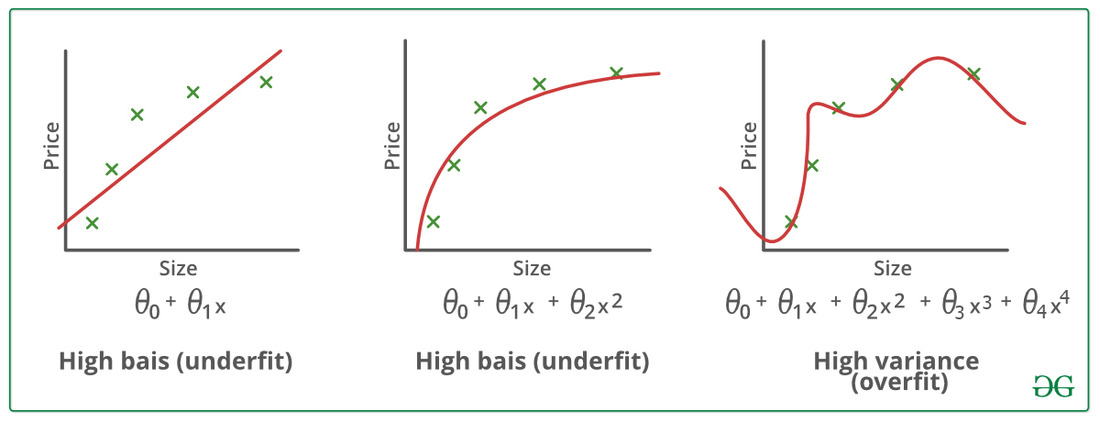
Overfitting: A statistical model is said to be overfitted when the model does not make accurate predictions on testing data. When a model gets trained with so much data, it starts learning from the noise and inaccurate data entries in our data set. And when testing with test data results in High variance. Then the model does not categorize the data correctly, because of too many details and noise. The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models. A solution to avoid overfitting is using a linear algorithm if we have linear data or using the parameters like the maximal depth if we are using decision trees.

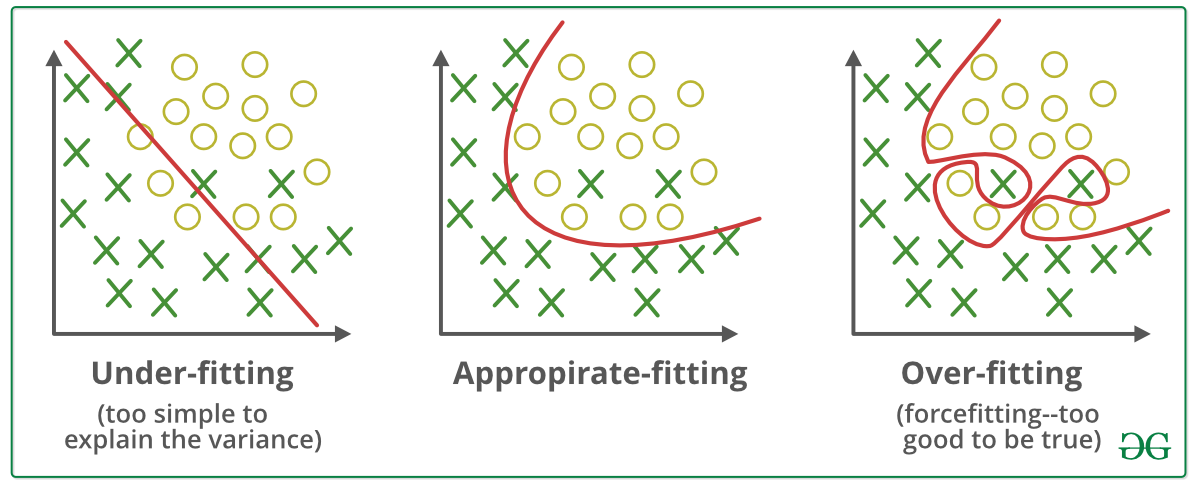
In a nutshell, Overfitting is a problem where the evaluation of machine learning algorithms on training data is different from unseen data.

Reasons for Overfitting are as follows:

* High variance and low bias
* The model is too complex
* The size of the training data

Examples:





Techniques to reduce overfitting:

* Increase training data.
* Reduce model complexity.
* Early stopping during the training phase (have an eye over the loss over the training period as soon as loss begins to increase stop training).
* Ridge Regularization and Lasso Regularization
* Use dropout for neural networks to tackle overfitting.

3. Bootstrapping vs. cross-validation

Cross validation splits the available dataset to create multiple datasets, and Bootstrapping method uses the original dataset to create multiple datasets after resampling with replacement. Bootstrapping is not as strong as Cross validation when it is used for model validation. Bootstrapping is more about building ensemble models or just estimating parameters.

Bootstrapping is any test or metric that relies on random sampling with replacement.It is a method that helps in many situations like validation of a predictive model performance, ensemble methods, estimation of bias and variance of the parameter of a model etc.

Cross validation is a procedure for validating a model's performance, and it is done by splitting the training data into k parts. We assume that the k-1 parts is the training set and use the other part is our test set. We can repeat that k times differently holding out a different part of the data every time. Finally, we take the average of the k scores as our performance estimation. Cross validation can suffer from bias or variance. Increasing the number of splits, the variance will increase too and the bias will decrease. On the other hand, if we decrease the number of splits, the bias will increase and the variance will decrease.

* Cross-Validation: provide estimates of the test error.
* Bootstrap: provides the standard error of the estimates.

10. Make quick notes on:

1. LOOCV.

Ans: Leave-one-out cross-validation, or LOOCV, is a configuration of k-fold cross-validation where k is set to the number of examples in the dataset.

LOOCV is an extreme version of k-fold cross-validation that has the maximum computational cost. It requires one model to be created and evaluated for each example in the training dataset.

The benefit of so many fit and evaluated models is a more robust estimate of model performance as each row of data is given an opportunity to represent the entirety of the test dataset.

Given the computational cost, LOOCV is not appropriate for very large datasets such as more than tens or hundreds of thousands of examples, or for models that are costly to fit, such as neural networks **Don’t Use LOOCV: Large datasets or costly models to fit.**

Given the improved estimate of model performance, LOOCV is appropriate when an accurate estimate of model performance is critical. This particular case when the dataset is small, such as less than thousands of examples, can lead to model overfitting during training and biased estimates of model performance.

Further, given that no sampling of the training dataset is used, this estimation procedure is deterministic, unlike train-test splits and other k-fold cross-validation confirmations that provide a stochastic estimate of model performance.

Use LOOCV: Small datasets or when estimated model performance is critical.

Once models have been evaluated using LOOCV and a final model and configuration chosen, a final model is then fit on all available data and used to make predictions on new data.

2. F-measurement:

In [statistical](https://en.wikipedia.org/wiki/Statistics) analysis of [binary classification](https://en.wikipedia.org/wiki/Binary_classification), the **F-score** or **F-measure** is a measure of a test's [accuracy](https://en.wikipedia.org/wiki/Accuracy_and_precision#In_binary_classification). It is calculated from the [precision](https://en.wikipedia.org/wiki/Precision_(information_retrieval)) and [recall](https://en.wikipedia.org/wiki/Recall_(information_retrieval)) of the test.



A more general F score, , that uses a positive real factor , where is chosen such that recall is considered times as important as precision, is:



## Applications[[edit](https://en.wikipedia.org/w/index.php?title=F-score&action=edit&section=6)]

The F-score is often used in the field of [information retrieval](https://en.wikipedia.org/wiki/Information_retrieval) for measuring [search](https://en.wikipedia.org/wiki/Web_search), [document classification](https://en.wikipedia.org/wiki/Document_classification), and [query classification](https://en.wikipedia.org/wiki/Query_classification) performance.[[15]](https://en.wikipedia.org/wiki/F-score#cite_note-15) Earlier works focused primarily on the F1 score, but with the proliferation of large scale search engines, performance goals changed to place more emphasis on either precision or recall[[16]](https://en.wikipedia.org/wiki/F-score#cite_note-16) and so  is seen in wide application.

The F-score is also used in [machine learning](https://en.wikipedia.org/wiki/Machine_learning).[[17]](https://en.wikipedia.org/wiki/F-score#cite_note-17) However, the F-measures do not take true negatives into account, hence measures such as the [Matthews correlation coefficient](https://en.wikipedia.org/wiki/Matthews_correlation_coefficient), [Informedness](https://en.wikipedia.org/wiki/Informedness) or [Cohen's kappa](https://en.wikipedia.org/wiki/Cohen%27s_kappa) may be preferred to assess the performance of a binary classifier.[[18]](https://en.wikipedia.org/wiki/F-score#cite_note-18)

The F-score has been widely used in the natural language processing literature,[[19]](https://en.wikipedia.org/wiki/F-score#cite_note-Derczynski2016-19) such as in the evaluation of [named entity recognition](https://en.wikipedia.org/wiki/Named_entity_recognition) and [word segmentation](https://en.wikipedia.org/wiki/Word_segmentation).

3. The width of the silhouette

Silhouette width is **a widely used index for assessing the fit of individual objects in the classification, as well as the quality of clusters and the entire classification**.Silhouette combines two clustering criteria, compactness and separation, which imply that spherical cluster shapes are preferred over others—a property that can be seen as a disadvantage in the presence of complex, nonspherical clusters, which is common in real situations.

### The original silhouette width

The original definition of silhouette width according to Rousseeuw ([1987](https://onlinelibrary.wiley.com/doi/full/10.1002/ece3.5774#ece35774-bib-0031)) is as follows. Let *i* be a focal object belonging to cluster *A*. Denote by *C* a cluster not containing *i*. *a*(*i*) is defined as the average dissimilarity between *i* and all other objects in *A*, while *c*(*i*,*C*) is the average dissimilarity between *i* and all objects in *C*.



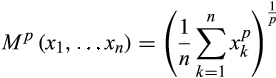
The silhouette width, *s*(*i*), is defined as:



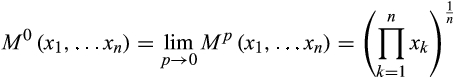
*s*(*i*) ranges between −1 and 1. Values near 1 indicate that object *i* is much closer to the other objects in the same cluster than to objects of the closest other cluster, implying a correct classification. If *s*(*i*) is near 0, the correct classification of the focal object is doubtful, thus suggesting an intermediate position between two clusters. *s*(*i*) near −1 indicates obvious misclassification. Accordingly, averaging silhouette widths over a cluster gives an assessment of the “goodness” of that cluster, or a sample-wise average can be used as an index of the validity of the entire classification. Instead of cluster-wise or sample-wise averages of *s*(*i*), the number or proportion of objects with positive silhouette width can also be used as validity measures. For a cluster containing a single object, *s*(*i*) takes the arbitrary value 0.

### Implementing the generalized mean

Applying the arithmetic mean to calculate average within- and between-cluster distances, as the index was introduced originally (Rousseeuw, [1987](https://onlinelibrary.wiley.com/doi/full/10.1002/ece3.5774#ece35774-bib-0031)), implies that the ideal cluster shape is spherical. However, this preference can be relaxed by choosing other types of means. Generalized mean (also called Hölder or power mean) offers a flexible solution to calculate sample means ranging between minimum and maximum (Cantrell & Weisstein, [2019](https://onlinelibrary.wiley.com/doi/full/10.1002/ece3.5774#ece35774-bib-0005)). Let *X* be a sample of positive real numbers *x*1, *x*2, …, *x*n and *p* an element of affinely extended real numbers. The generalized mean of degree *p* is as follows:



For *p* = 0 and *p* = |∞| the following exceptions are to be made:







The generalized mean takes the values of well-known summary statistics presented in Table [1](https://onlinelibrary.wiley.com/doi/full/10.1002/ece3.5774#ece35774-tbl-0001). The original version of silhouette width is the special case when within- and between-group average distances are calculated by *p* = 1. By changing the *p* parameter, it is possible to emphasize lower or higher distances in the calculation of means. The lower the *p* parameter is, the more importance is attributed to objects in close proximity, while the effect of farther neighbor objects (including outliers) is reduced. In this way, the criteria of compactness are gradually replaced by connectedness and clusters with irregular or elongated shape can also be considered “good”. At *p* = −∞, a classification is ideal if each object is assigned to the same cluster as the most similar other object in the sample. This procedure follows the logic of single linkage clustering, while the original version making use of arithmetic averages followed the logic of average linkage. In contrast, when *p* > 1, the compactness criterion is attributed higher weight; thus, the preference toward spherical clusters is further increased and the effect of outliers on the overall classification should become more significant. At *p* = +∞, the clustering criteria of complete linkage are applied.

Table 1. Special cases of the generalized mean

| ***p*** | **Descriptive statistic** |
| --- | --- |
| −∞ | Minimum |
| −1 | Harmonic mean |
| 0 | Geometric mean |
| 1 | Arithmetic mean |
| 2 | Quadratic mean (root-mean-square) |
| +∞ | Maximum |

4. Receiver operating characteristic curve

A **receiver operating characteristic curve**, or **ROC curve**, is a [graphical plot](https://en.wikipedia.org/wiki/Graph_of_a_function) that illustrates the diagnostic ability of a [binary classifier](https://en.wikipedia.org/wiki/Binary_classifier) system as its discrimination threshold is varied

The ROC curve is created by plotting the [true positive rate](https://en.wikipedia.org/wiki/True_positive_rate) (TPR) against the [false positive rate](https://en.wikipedia.org/wiki/False_positive_rate) (FPR) at various threshold settings. The true-positive rate is also known as [sensitivity](https://en.wikipedia.org/wiki/Sensitivity_(tests)), [recall](https://en.wikipedia.org/wiki/Precision_and_recall#Definition_(classification_context)) or *probability of detection*.[[10]](https://en.wikipedia.org/wiki/Receiver_operating_characteristic#cite_note-matlab-10) The false-positive rate is also known as *probability of false alarm*[[10]](https://en.wikipedia.org/wiki/Receiver_operating_characteristic#cite_note-matlab-10) and can be calculated as (1 − [specificity](https://en.wikipedia.org/wiki/Specificity_(tests))).The ROC can also be thought of as a plot of the [power](https://en.wikipedia.org/wiki/Statistical_power) as a function of the [Type I Error](https://en.wikipedia.org/wiki/Type_I_Error) of the decision rule (when the performance is calculated from just a sample of the population, it can be thought of as estimators of these quantities). The ROC curve is thus the sensitivity or recall as a function of [fall-out](https://en.wikipedia.org/wiki/False_positive_rate). In general, if the probability distributions for both detection and false alarm are known, the ROC curve can be generated by plotting the [cumulative distribution function](https://en.wikipedia.org/wiki/Cumulative_distribution_function) (area under the probability distribution from  to the discrimination threshold) of the detection probability in the y-axis versus the cumulative distribution function of the false-alarm probability on the x-axis.

ROC analysis provides tools to select possibly optimal models and to discard suboptimal ones independently from (and prior to specifying) the cost context or the class distribution. ROC analysis is related in a direct and natural way to cost/benefit analysis of diagnostic [decision making](https://en.wikipedia.org/wiki/Decision_making).

The ROC is also known as a relative operating characteristic curve, because it is a comparison of two operating characteristics (TPR and FPR) as the criterion changes

ROC Space:

To draw a ROC curve, only the true positive rate (TPR) and false positive rate (FPR) are needed (as functions of some classifier parameter). The TPR defines how many correct positive results occur among all positive samples available during the test. FPR, on the other hand, defines how many incorrect positive results occur among all negative samples available during the test.

A ROC space is defined by FPR and TPR as *x* and *y* axes, respectively, which depicts relative trade-offs between true positive (benefits) and false positive (costs). Since TPR is equivalent to sensitivity and FPR is equal to 1 − specificity, the ROC graph is sometimes called the sensitivity vs (1 − specificity) plot. Each prediction result or instance of a [confusion matrix](https://en.wikipedia.org/wiki/Confusion_matrix) represents one point in the ROC space.

The best possible prediction method would yield a point in the upper left corner or coordinate (0,1) of the ROC space, representing 100% sensitivity (no false negatives) and 100% specificity (no false positives). The (0,1) point is also called a *perfect classification*. A random guess would give a point along a diagonal line (the so-called *line of no-discrimination*) from the bottom left to the top right corners (regardless of the positive and negative [base rates](https://en.wikipedia.org/wiki/Base_rate)).[[25]](https://en.wikipedia.org/wiki/Receiver_operating_characteristic#cite_note-25) An intuitive example of random guessing is a decision by flipping coins. As the size of the sample increases, a random classifier's ROC point tends towards the diagonal line. In the case of a balanced coin, it will tend to the point (0.5, 0.5).

The diagonal divides the ROC space. Points above the diagonal represent good classification results (better than random); points below the line represent bad results (worse than random). Note that the output of a consistently bad predictor could simply be inverted to obtain a good predictor.

|  |  |
| --- | --- |