**CHAPTER 7: Resampling, Evaluation, and Hyperparameter Tuning**

## Introduction

Whenever we build any machine learning model, we feed it with initial data to rain the model. And then we feed some unknown data (test data) to understand how well the model performs and generalized over unseen data. If the model performs well on the unseen data, it’s consistent, and can predict with good accuracy on a wide range of input data; then this model is stable.

We're going to look at four different techniques that we can use to split our training dataset and create useful estimates of performance for our Machine learning algorithms.

* Train and Test Sets.
* K-fold cross-validation
* Leave One Out Cross-Validation
* Stratified K-Fold Cross-Validation

## Split into Train and Test sets

A key step in process of building a machine learning model is training and testing the dataset. Without training the algorithm, the model parameters do not converge on useful values, and the model is useless. Without testing the algorithm after training the model parameters, there is no way to know how good the model actually is, which is equally useless.

The train test split technique can be used for classification regression problems to test machine learning algorithms. The procedure takes the given dataset and splits it into two subjects:

* Training dataset: it is used to train the algorithm and fit the machine learning model.
* Test dataset: Using the input element from the training data, the algorithms make predictions.

The model is first to fit on the available data with known inputs and outputs. It is then run to make predictions on the rest of the data subset to learn from it. This can be used to make predictions on future data sets where the expected input and output values are non-existent.

You can use the scikit learn machine learning library play in Python for the implementation of the train test split valuation procedure using the train\_test\_split() function. it works by taking the dataset loaded as the input and outputs it split as two subsets.



Here,

X: Independent Features

Y: Dependent Feature

Test\_size: The ratio in which split needs to be done.

## Cross-Validation

Suppose you train a model on a given dataset using any specific algorithm. You tried to find the accuracy of the trained model using the same training data and found the accuracy to be 95% or maybe even 100%. What does this mean? Is your model ready for prediction? The answer is no. Why? Because your model has trained itself on the given data, i.e. it knows the data and it has generalized over it very well. But when you try and predict over a new set of data, it’s most likely to give you very bad accuracy, because it has never seen the data before and thus it fails to generalize well over it. This is the problem of overfitting. To tackle such a problem, Cross-validation comes into the picture. Cross-validation is a resampling technique with a basic idea of dividing the training dataset into two parts i.e. train and test. On one part(train) you try to train the model and on the second part(test) i.e. the data which is unseen for the model, you make the prediction and check how well your model works on it. If the model works with good accuracy on your test data, it means that the model has not overfitted the training data and can be trusted with the prediction, whereas if it performs with bad accuracy then our model is not to be trusted we need to tweak our algorithm.

* **Hold Out Method**

It is the most basic of the CV techniques. It simply divides the dataset into two sets of training and test. The training dataset is used to train the model and then test data is fitted in the trained model to make predictions. We check the accuracy and assess our model on that basis. This method is used as it is computationally less costly. But the evaluation based on the Hold-out set can have a high variance because it depends heavily on which data points end up in the training set and which in test data. The evaluation will be different every time this division changes.

* **K-Fold Cross-Validation**

**Chart, diagram

Description automatically generated**

To tackle the high variance of the Hold-out method, the k-fold method is used. The idea is simple, divide the whole dataset into ‘k’ sets preferably of equal sizes. Then the first set is selected as the test set and the rest ‘k-1’ sets are used to train the data. Error is calculated for this particular dataset. Then the steps are repeated, i.e., the second set is selected as the test data, and the remaining ‘k-1’ sets are used as the training data. Again, the error is calculated. Similarly, the process continues for ‘k’ times. In the end, the CV error is given as the mean of the total errors calculated individually, mathematically given as:

**A picture containing text, clock, watch

Description automatically generated**

The variance in error decreases with the increase in ‘k’. The disadvantage of k-fold cv is that it is computationally expensive as the algorithm runs from scratch for ‘k’ times.

* **Leave One Out Cross Validation (LOOCV)**

**Chart

Description automatically generated with low confidence**

LOOCV is a special case of k-fold CV, where k becomes equal to n (number of observations). So instead of creating two subsets, it selects a single observation as test data and the rest of the data as the training data. The error is calculated for these test observations. Now, the second observation is selected as test data, and the rest of the data is used as the training set. Again, the error is calculated for this particular test observation. This process continues ‘n’ times and in the end, CV error is calculated as:

Diagram, schematic

Description automatically generated

* **Stratified K-Fold Cross-Validation**

This is a slight variation from K-Fold Cross Validation, which uses ‘stratified sampling’ instead of ‘random sampling’.

Let’s quickly understand what stratified sampling is and how is it different from random sampling.

Suppose your data contains reviews for a cosmetic product used by both the male and female population. When we perform random sampling to split the data into train and test sets, there is a possibility that most of the data representing males is not represented in training data but might end up in test data. When we train the model on sample training data that is not a correct representation of the actual population, the model will not predict the test data with good accuracy.

This is where Stratified Sampling comes to the rescue. Here the data is split in such a way that it represents all the classes from the population.

A picture containing diagram

Description automatically generated

This is exactly what stratified K-Fold CV does and it will create K-Folds by preserving the percentage of sample for each class. This solves the problem of random sampling associated with Hold out and K-Fold methods.

* **Repeated Random Train-Test Splits**

This technique is a hybrid of traditional train-test splitting and the k-fold cross-validation method. In this technique, we create random splits of the data in the training-test set manner and then repeat the process of splitting and evaluating the algorithm multiple times, just like the cross-validation method.

## Bias - Variance Trade-off

There is a very good explanation given in the ISLR Book as given below:

A k-fold CV with k < n has a computational advantage to LOOCV. But putting computational issues aside, a less obvious but potentially more important advantage of k-fold CV is that it often gives more accurate estimates of the test error rate than does LOOCV. The validation set approach can lead to overestimates of the test error rate since in this approach the training set used to fit the statistical learning method contains only half the observations of the entire data set. Using this logic, it is not hard to see that LOOCV will give approximately unbiased estimates of the test error since each training set contains n − 1 observation, which is almost as many as the number of observations in the full data set. And performing k-fold CV for, say, k = 5 or k = 10 will lead to an intermediate level of bias since each training set contains (k − 1)n/k observations—fewer than in the LOOCV approach, but substantially more than in the validation set approach. Therefore, from the perspective of bias reduction, it is clear that LOOCV is to be preferred to k-fold CV. However, we know that bias is not the only source for concern in an estimating procedure; we must also consider the procedure’s variance. It turns out that LOOCV has a higher variance than does k-fold CV with k < n. Why is this the case? When we perform LOOCV, we are in effect averaging the outputs of n fitted models, each of which is trained on an almost identical set of observations; therefore, these outputs are highly (positively) correlated with each other. In contrast, when we perform k-fold CV with k < n, we are averaging the outputs of k fitted models that are somewhat less correlated with each other since the overlap between the training sets in each model is smaller. Since the mean of many highly correlated quantities has higher variance than does the mean of many quantities that are not as highly correlated, the test error estimate resulting from LOOCV tends to have higher variance than does the test error estimate resulting from k-fold CV.

**Best Practices and tips**

It’s worth mentioning that sometimes performing cross-validation might be a little tricky.

For example, it’s quite easy to make a logical mistake when splitting the dataset which may lead to an untrustworthy CV result.

You may find some tips that you need to keep in mind when cross-validating a model below:

* Be logical when splitting the data
* Use the proper CV method
* When working with time series don’t validate on the post
* When working with medical or financial data remember to split by person. Avoid having data for one person both in the training and test set as it may be considered a data leak.
* When cropping patches from larger images remember to split by the large image id.

**What Technique to use when**

A cross-Validation is a powerful tool. Every data scientist should be familiar with it. In real life, you can’t finish the project without cross-validating a model.

In my opinion, the best CV techniques are K-Fold and Stratified K-fold. Personally, I used them in the task of Fraud Detection.

**Practical Implementation**

Let’s see the implementation of the cross-validation techniques that we discussed.

**Importing the data**

Graphical user interface, text, application

Description automatically generated

Graphical user interface

Description automatically generated

**Separating the Independent and Dependent Features**

A picture containing company name

Description automatically generated

Graphical user interface

Description automatically generated with medium confidence

**Dropping the Null Values**



**Hold Out Validation Approach – Train and Test Split**

Graphical user interface, text, application

Description automatically generated

The model score using the Hold Out approach is 90%. Let’s go and implement other techniques and see the performance.

**K Fold Cross-Validation**

Text

Description automatically generated

Text

Description automatically generated

The mean square using K-Fold cross-validation is 92%. 2% increase in the score compared to the Hold Out technique.

**Stratified K-fold Cross-Validation**

Graphical user interface, text

Description automatically generated

The mean score using Stratified K-fold cross-validation is 92%. There is not much improvement in the score.

**Leave One Out Cross Validation (LOOCV)**

Text

Description automatically generated

There is a decimal point increase in the mean score by using the LOOCV technique.

Now, let’s look at our final technique

**Repeated Random Test-Train Splits**

Text

Description automatically generated

Graphical user interface, text, application

Description automatically generated

The performance of the Repeated Random technique is almost similar to the other cross-validation technique.

So, most of the random samples were able to give almost similar results. You can go ahead use any of the following techniques and build the model.

## Evaluation Metrics

Evaluation metrics are tied to machine learning tasks. There are different metrics for the tasks of classification and regression. Some metrics, like precision-recall, are useful for multiple tasks. classification and regression are examples of supervised learning, which constitute most machine learning applications. using different metrics for performance evaluation, we should be able to improve our model’s overall predictive power before we roll it out for production on unseen data. Without doing a proper evaluation of the machine learning model by using different evaluation metrics, and only depending on accuracy, can lead to a problem when the respective model is deployed on unseen data and may end in poor predictions.

## Model Evaluation Metrics for Regression

Model evaluation is very important in data analytics. It helps you to understand the performance of your model and makes it easy to present your model to other people. There are many different evaluation metrics out there but only some of them are suitable to be used for regression.

There are 5 main metrics for model evaluation in regression:

1. R Square
2. Adjusted R Square
3. Mean Square Error (MSE)
4. Root Mean Squared Error (RMSE)
5. Mean Absolute Error
6. **R Square/Adjusted R Square**

R Square measures how much variability in the dependent variables can be explained by the model. It is a square of Correlation Coefficient (R) and that is why it is called R Square. It takes the form of a proportion --- the proportion of variance explained --- and so it always takes on a value between 0 and 1.

For example, **R2** statistic = 0.75, says that our model fits 75% of the total dataset. Similarly, if it is 0, it means none of the data points is being explained and a value of 1 represents 100% data explanation. Mathematically **R2** statistic is calculated as:

Text

Description automatically generated

Where RSS: is the Residual Sum of Squares and is given as:

A picture containing text, watch

Description automatically generated

RSS is the residual (error) term we have been talking about so far. And. TSS: is the Total Sum of Squared and given as:



TSS is calculated when we consider the line passing through the mean value y. to be the best fit line. Just like RSS, we calculate the error term when the best fit line is the line passing through the mean value of y and we get the value of TSS.

A picture containing timeline

Description automatically generated

The closer the value of R2 is to 1 the better the model fits our data. If R2 comes below (which is the possibility) that means the model is so bad that it is performing even worse than the average best fit.

1. **Adjusted R Square**

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. That is why Adjusted R Square is introduced because it will penalize additional independent variables added to the model and adjust the metric to prevent Overfitting issues.

Mathematically it is calculated as:

A picture containing diagram

Description automatically generated

Where:

* R2: Sample R Square
* N: Total Sample Size
* P: Number of Independent variables

1. **Mean Square Error (MSE)**

While R Square is a relative measure of how well the model fits dependent variables, Mean Square Error is an absolute measure of the goodness of fit.

MSE is calculated by the sum of prediction error which is real output minus predicted output and then divided by the number of data points. It gives you an absolute number on how much you’re predicted results deviate from the actual number. You cannot interpret much insight from one single result but it gives you a real number to compare against other model results and help you to select the best regression model.

Mathematically it is calculated as:

A picture containing text, clock, gauge

Description automatically generated

Where:

* Yi: is an actual value
* Yi hat: is predicted value
* N: No. of observations

1. **Root Mean Square Error (RMSE)**

Root Mean Square Error (RMSE) is the square root of MSE. It is used more commonly than MSE because sometimes MSE values can be too big to compare easily. Secondly, MSE is calculated by the square of error, and thus square root brings to the same level of prediction error and makes it easier for interpretation.

Mathematically is calculated as:

Text

Description automatically generated

Where:

* Yi: is an actual value
* Yi hat: is predicted value
* N: No. of observations

1. **Mean Absolute Error**

Mean Absolute Error (MAE) is like Mean Square Error (MSE). However, instead of the sum of square error in MSE, MAE is taking the Sum of Absolute value of error.

Compared to MSE, RMSE, and MAE, MAE is a more direct representation of the sum of error terms. MSE gives larger penalization to big prediction errors by squaring them while MAE treats all errors the same.

Mathematically it is calculated as:

A picture containing text, clock, gauge

Description automatically generated

Where:

* Yi: is an actual value
* Yi hat: is predicted value
* N: No. of observations

**Overall Recommendation/Conclusion**

R Square/Adjusted R Square is better used to explain the model to other people because you can explain the number as a percentage of the output variability. MSE, RMSE, or MAE is better to be used to compare performance between different regression models.

## Evaluation of Classification Model

In machine learning, once we have a result of the classification problem, how do we measure how accurate our classification is?

In a Classification problem, the credibility of the model is measured using the confusion matrix generated, i.e., how accurately the true positive and true negatives were predicted.

The different metrics used for this purpose are:

* Accuracy
* Recall
* Precision
* F1 score
* Specificity or True Negative Rate
* AUC (Area under the curve)
* RUC (Receiver Operator Characteristic)
* **Confusion Matrix**

A typical confusion matrix looks like the figure shown.

Table

Description automatically generated

Where the terms have the meaning:

* **True Positive (TP):** A result that was predicted as positive by the classification model and also is positive.
* **True Negative (TN):** A result that was predicted as negative by the classification model and also is negative.
* **False Positive (FP):** A result that was predicted as positive by the classification model but is negative.
* **False Negative (FN):** A result that was predicted as negative by the classification model but concerning the ability of the model is based on how many correct predictions the model makes.

**Accuracy**

The mathematical formula is:

A picture containing shape

Description automatically generated

It can be said that it’s defined as the total number of correct classifications divided by the total number of classifications.

**Recall or Sensitivity**

The mathematical formula is:

Text

Description automatically generated with medium confidence

As the name suggests, it is a measure of: from the total number of positive results how many positives were correctly predicted by the model.

It shows how relative the model is, in terms of positive results only.

Let’s suppose in the previous model, the model made 50 correct predictions (TP) but failed to identify 200 cancer patients (FN). Recall in that case will be:

**= 0.2**

The model was able to recall only 20% of the cancer patients

**Precision**

Precision is a measure of amongst all the positive predictions, how many of them were actually positive. Mathematically,



Let’s suppose in the previous example,

The model identified 50 people as cancer patients (TP) but also raised a false alarm for 100 (FP).

**= 0.33**

The model only has a precision of 33%

**But we have a problem!!**

As evident from the previous example, the model had a very high Accuracy but performed poorly in terms of Precision and Recall. So, necessarily *Accuracy* is not the metric to use for evaluating the model in this case.

Imagine a scenario, where the requirement was that the model recalled all the defaulters who did not pay back the loan. Suppose there were 10 such defaulters and to recall those 10 defaulters, and the model gave you 20 results out of which only the 10 are the actual defaulters. Now, the recall of the model is 100%, but the precision goes down to 50%.

**Trade-off?**

Chart, histogram

Description automatically generated

As observed from the graph, with an increase in the Recall, there is a drop in Precision of the model.

So the question is - what to go for? Precision or Recall?

Well, the answer is: it depends on the business requirement.

For example, if you are predicting cancer, you need a 100 % recall. But suppose you are predicting whether a person is innocent or not, you need 100% precision.

Can we maximize both at the same time? No

So, there is a need for a better metric then?

Yes. And it’s called an *F1 Score*

**F1 Score**

From the previous examples, it is clear that we need a metric that considers both Precision and Recall for evaluating a model. One such metric is the F1 score.

F1 score is defined as the harmonic mean of Precision and Recall.

The mathematical formula is:

**F1 Score** =

**Specificity or True Negative Rate**

This represents how specific is the model while predicting the True Negatives. Mathematically,

**Specificity** =

It can be said that it quantifies the total number of negatives predicted by the model concerning the total number of actual negative or non-favorable outcomes.

Similarly, False Positive rate can be defined as (1 – Specificity) or

**ROC (Receiver Operator Characteristic)**

We know that the classification algorithms work on the concept of probability of occurrence of the possible outcomes. A probability value lies between 0 and 1. Zero means that there is no probability of occurrence and one means that the occurrence is certain.

But while working with real-time data, it has been observed that we seldom get a perfect 0 or 1 value. Instead of that, we get different decimal values lying between 0 and 1. Now the question is if we are not getting binary probability values determining the class in our classification problem?

There comes the concept of Threshold. A threshold is set, any probability value below the threshold is a negative outcome, and anything more than the threshold is a favorable or positive outcome. For Example, if the threshold is 0.5, any probability value below 0.5 means a negative or an unfavorable outcome and any value above 0.5 indicates a positive or favorable outcome.

Now, the question is what should be an ideal threshold?

The following diagram shows a typical logistic regression curve.

Chart, line chart

Description automatically generated

* The horizontal lines represent the various values of thresholds ranging from 0 to 1.
* Let’s suppose our classification problem was to identify the obese people from the given data.
* The green markers represent obese people and the red markers represent non-obese people.
* Our confusion matrix will depend on the value of the threshold chosen by us.
* For Example, if 0.25 is the threshold then
* TP (Actually obese) = 3
* TN (Not obese) = 2
* FP (Not obese but predicted obese) = 2 (The two red squares above the 0.25 line)
* FN (Obese but predicted as not obese) = 1 (The Green circle below 0.25 line)

A typical ROC curve looks like the following figure.

Chart, line chart

Description automatically generated

* Mathematically, it represents the various confusion matrices for various thresholds. Each black dot is one confusion matrix.
* The green dotted line represents the scenario when the true positive rate equals the false positive rate.
* As evident from the curve, as we move from the rightmost dot towards the left, after a certain threshold, the false positive rate decreases.
* After some time, the false positive rate becomes zero.
* The point encircled in green is the best point as it predicts all the values correctly and keeps the false positive as a minimum.
* But that is not a rule of thumb. Based on the requirement, we need to select the point of a threshold.
* The ROC curve answers our question of which threshold to choose.

**But we have confusion!!**

Let’s suppose that we used different classification algorithms, and different ROCs for the corresponding algorithms have been plotted. The question is: which algorithm to choose now? The answer is to calculate the area under each ROC curve.

**Area Under Curve (AUC)**

Chart, line chart

Description automatically generated

* It helps us to choose the best model amongst the models for which we have plotted the ROC curves
* The best model is the one that encompasses the maximum area under it.
* In the adjacent diagram, amongst the two curves, the model that resulted in the red one should be chosen as it has the blue one.

## Hyperparameter Tuning

Hyperparameter tuning is choosing a set of optimal hyperparameters for a learning algorithm. A hyperparameter is a model argument whose value is set before the learning processing begins. The key to machine learning algorithms is hyperparameter tuning.

**Hyperparameter types:**

* **Bias:** error due to simplifying model assumptions
* **Variance:** error due to randomness of the training set.

The trade-off between these components is determined by the complexity of the model and the amount of training data. The optimal hyperparameters help to avoid under-fitting (training and test set and are both high) and overfitting (training error is low but test error is high).

**Hyper-parameters:** Model parameters are learned from data and hyperparameters are tuned to get the best fit. Searching for the best hyper-parameter can be tedious, hence search algorithms like grid search and random search are used.

A screenshot of a computer

Description automatically generated with low confidence

Model selection refers to the process of selecting the right model that fits the data. This is done using test evaluation metrics. The results from the test data are passed back to the hyper-parameter tuner to get the most optimal hyperparameters.

Diagram

Description automatically generated

We shall look into two of the most widely-used parameter optimiser techniques

* Grid Search
* Random Search

## Grid Search

Grid search is a technique which tends to find the right set of hyperparameters for the particular model. Hyperparameters are not the model parameters, and it is not possible to find the best set from the training data. Model parameters are learned during training when we optimise a loss function using something like a gradient descent. In this tuning technique, we simply build a model for every combination of various hyperparameters and evaluate each model. The model which gives the highest accuracy wins. The pattern followed here is similar to the grid, where all the values are placed in the form of a matrix. Each set of parameters is taken into consideration and the accuracy is noted. Once all the combinations are evaluated, the model with the set of parameters which give the top accuracy is considered to be the best. Below is a visual description of uniform search pattern of the grid search.

A picture containing text

Description automatically generated

One of the drawbacks of grid search is that when it comes to dimensionality, it suffers when evaluating the number of hyperparameters grows exponentially. However, there is no guarantee that the search will produce the perfect solution, as it usually finds one by aliasing around the right set.

## Random Search

Random search is a technique where random combinations of the hyperparameters are used to find the best solution for the built model. It is similar to grid search, and yet it has proven to yield better results comparatively. The drawback of random search is that it yields high variance during computing. Since the selection of parameters is completely random; and since no intelligence is used to sample these combinations, luck plays its part.

Diagram

Description automatically generated

As random values are selected at each instance, it is highly likely that the whole of action space has been reached because of the randomness, which takes a huge amount of time to cover every aspect of the combination during grid search. This works best under the assumption that not all hyperparameters are equally important. In this search pattern, random combinations of parameters are considered in every iteration. The chances of finding the optimal parameter are comparatively higher in random search because of the random search pattern where the model might end up being trained on the optimised parameters without any aliasing.

## Hands-on

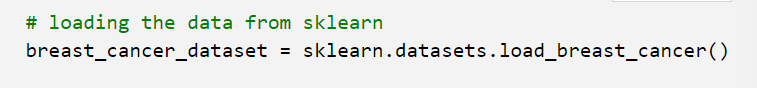
**Step1:** Importing the required libraries

Text

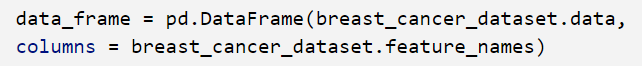
Description automatically generated

We will be working on the breast cancer dataset

**Step2:** Loading the data



Load the data to a data frame



Table

Description automatically generated

adding the 'target' column to the data frame

A picture containing shape

Description automatically generated

Print the first 5 rows of the data frame



Table

Description automatically generated

**Step3:** Checking the null values

A picture containing company name

Description automatically generated

There are no missing values in the data.

**Step4:** Checking the distribution of the Target Variable

Chart, waterfall chart

Description automatically generated

Graphical user interface, application

Description automatically generated

**Step5:** Separating the features and target

Chart

Description automatically generated

**Step6:** GridSearchCV

GridSearchCV is used for determining the best parameters for our model. We will be using support vector machine classifier to demonstrate.

loading the SVC model

A picture containing text

Description automatically generated

hyperparameters

A picture containing chart

Description automatically generated

Grid search



The “cv” parameter here is nothing but the cross validation.

Fitting the data to our model

A picture containing text

Description automatically generated

A picture containing diagram

Description automatically generated

A picture containing company name

Description automatically generated

Text, letter

Description automatically generated

A screenshot of a computer

Description automatically generated with low confidence

best parameters can be obtained from the above classifier.

Text

Description automatically generated

A picture containing text

Description automatically generated

The best parameters from the grid search cv are c=10, kernel = ‘linear’. We can check the best score obtained from the best parameters.

Text

Description automatically generated



### Randomized Search CV

Loading the SVC model



hyperparameters

A picture containing chart

Description automatically generated

Random Search



Fitting the data to our model

Graphical user interface, text, application, email

Description automatically generated

A picture containing logo

Description automatically generated

A black and white document

Description automatically generated with low confidence

Text

Description automatically generated

Best parameters

Text

Description automatically generated

The best parameters obtained from random search cv are c=10, and kernel = ‘linear’. Let’s check the accuracy produced from the best parameters.

Text, letter

Description automatically generated

We got accuracy 95% using both the techniques. We can go ahead and build the model using the best parameters.

**SUMMARY**

In this chapter, you discovered statistical techniques that you can use to estimate the performance of your machine learning algorithms, called resampling. Specially, you learned about Cross Validation Techniques. Then, you learned how you can evaluate the performance of Regression and Classification Algorithms using a suite of different metrics, and finally we have seen how the tune our model hyperparameters using grid search and random search.

## Assessment

**Choose the appropriate option**

1. Given 100% of data is used for training, the validation technique can be called as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   1. Hold Out
   2. K-fold cross-validation
   3. LOOCV
   4. Resubstitution
2. Given 80% of data is selected for training and remaining 20% for testing, this validation technique can be called as \_\_\_\_\_\_\_\_\_\_\_\_\_\_
   1. Hold Out
   2. K-fold cross-validation
   3. LOOCV
   4. Resubstitution
3. Given 80% of data is selected for training and the remaining 20% for testing, and this process is carried out four times and the error rate is averaged out, this validation technique can be called as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   1. Hold Out
   2. K-fold cross-validation
   3. LOOCV
   4. Resubstitution
4. In the K-fold cross-validation technique, the value of k being large could lead to which of the following in relation to error rate.
   1. High Bias
   2. High Variance
   3. Low Bias, Large Variance
   4. Large Bias, Low Variance
5. The most common choice for K in K-fold cross validation technique is \_\_\_\_\_\_\_\_\_\_\_\_
   1. 3
   2. 5
   3. 10
   4. 12

**Fill in the spaces with appropriate answers**

1. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ refers to the ratio of negative correctly predicted from all the false labels.
2. \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ is the proportion of true positives out of predicted positives.
3. Given 1000 records, 1000 models are trained with 99 records as part of the training sample and remaining 1 sample for testing, and the error rate is average out, this validation technique can be called as \_\_\_\_\_\_\_\_\_\_\_\_\_
4. The process of making sure that there is an equal split of classes in training and test samples are called as \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
5. In K-fold cross validation technique, the value of k being small could lead to \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ relation to error rate.

**True or False**

1. Accuracy is not the best metric for classification problem statements.
   1. True
   2. False
2. ROC helps us to choose the best model amongst the models for which we have plotted the ROC curves.
   1. True
   2. False
3. For N records, LOOCV can also be called the N-fold cross-validation
   1. Ture
   2. False
4. In the K-fold cross-validation technique, the value of k being small could lead to overfitting.
   1. True
   2. False
5. Sensitivity = TP/(TP + FN)
   1. True
   2. False

## Programming Assessment

Using the data in the below URL, Perform the following tasks

https://github.com/fenago/MLBook/blob/main/Chapter%206%20-%20Classification/Code/Dataset/Cars.txt

1. Import the data
2. Perform Data Cleaning
3. Perform EDA
4. Fit Decision Tree, KNN, SVC, and Random Forest Classification models.
5. Perform Hyperparameter tuning using Grid Search CV and Random Search CV
6. Evaluate the models.

## Assessment Solutions

**Choose the appropriate option**

1. D
2. A
3. B
4. C
5. C

**Fill in the spaces with appropriate answers**

1. Ture Negative Rate
2. Precision
3. LOOCV
4. Stratification
5. Low bias, Large Variance

**True or False**

1. True
2. True
3. True
4. False
5. False