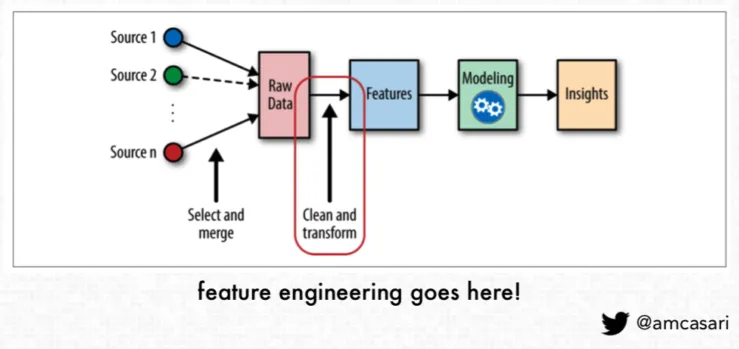
**1. What is feature engineering, and how does it work? Explain the various aspects of feature engineering in depth.**

Ans: Feature engineering is the pre-processing step of machine learning, which is used to transform raw data into features that can be used for creating a predictive model using Machine learning or statistical Modelling. Feature engineering in machine learning aims to improve the performance of models.

It does by **selecting, manipulating, and transforming raw data into features .**

****

Some common types of feature engineering include:

* Scaling and normalization means adjusting the range and center of data to ease learning and improve the interpretation of the results.
* Filling missing values implies filling in null values based on expert knowledge, heuristics, or by some machine learning techniques. Real-world datasets can be missing values due to the difficulty of collecting complete datasets and because of errors in the data collection process.
* Feature selection means removing features because they are unimportant, redundant, or outright counterproductive to learning. Sometimes you simply have too many features and need fewer.
* Feature coding involves choosing a set of symbolic values to represent different categories. Concepts can be captured with a single column that comprises multiple values, or they can be captured with multiple columns, each of which represents a single value and has a true or false in each field. For example, feature coding can indicate whether a particular row of data was collected on a holiday. This is a form of feature construction.
* Feature construction creates a new feature(s) from one or more other features. For example, using the date you can add a feature that indicates the day of the week. With this added insight, the algorithm could discover that certain outcomes are more likely on a Monday or a weekend.
* Feature extraction means moving from low-level features that are unsuitable for learning — practically speaking, you get poor testing results — to higher-level features that are useful for learning. Often feature extraction is valuable when you have specific data formats — like images or text — that have to be converted to a tabular row-column, example-feature format.

**2. What is feature selection, and how does it work? What is the aim of it? What are the various methods of function selection?**

Ans:

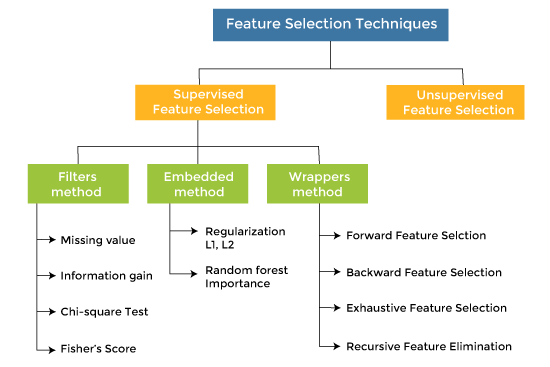
Feature selection is about selecting the subset of the original feature set,

AIM : The goal of feature selection techniques in machine learning is to find the best set of features that allows one to build useful models of studied phenomena.

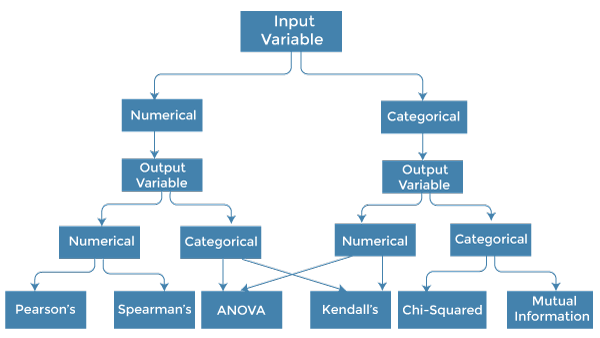
METHODS:

Supervised Techniques: Supervised Feature selection techniques consider the target variable and can be used for the labelled dataset.

Unsupervised Techniques: Unsupervised Feature selection techniques ignore the target variable and can be used for the unlabelled dataset.



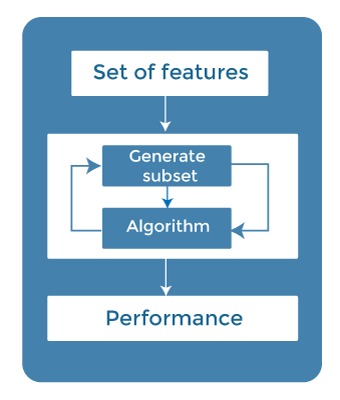
## How to choose a Feature Selection Method?



**3. Describe the function selection filter and wrapper approaches. State the pros and cons of each approach?**

Ans: 1. Wrapper Methods

In wrapper methodology, selection of features is done by considering it as a search problem, in which different combinations are made, evaluated, and compared with other combinations. It trains the algorithm by using the subset of features iteratively.



On the basis of the output of the model, features are added or subtracted, and with this feature set, the model has trained again.

Some techniques of wrapper methods are:

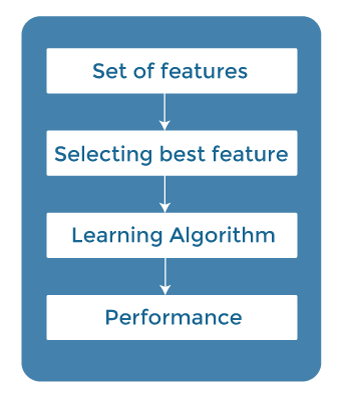
* **Forward selection** - Forward selection is an iterative process, which begins with an empty set of features. After each iteration, it keeps adding on a feature and evaluates the performance to check whether it is improving the performance or not. The process continues until the addition of a new variable/feature does not improve the performance of the model.
* **Backward elimination** - Backward elimination is also an iterative approach, but it is the opposite of forward selection. This technique begins the process by considering all the features and removes the least significant feature. This elimination process continues until removing the features does not improve the performance of the model.
* **Exhaustive Feature Selection-** Exhaustive feature selection is one of the best feature selection methods, which evaluates each feature set as brute-force. It means this method tries & make each possible combination of features and return the best performing feature set.
* **Recursive Feature Elimination-**Recursive feature elimination is a recursive greedy optimization approach, where features are selected by recursively taking a smaller and smaller subset of features. Now, an estimator is trained with each set of features, and the importance of each feature is determined using *coef\_attribute* or through a *feature\_importances\_attribute.*

### 2. Filter Methods

In Filter Method, features are selected on the basis of statistics measures. This method does not depend on the learning algorithm and chooses the features as a pre-processing step.

The filter method filters out the irrelevant feature and redundant columns from the model by using different metrics through ranking.

The advantage of using filter methods is that it needs low computational time and does not overfit the data.



Some common techniques of Filter methods are as follows:

* Information Gain
* Chi-square Test
* Fisher's Score
* Missing Value Ratio

**Information Gain:** Information gain determines the reduction in entropy while transforming the dataset. It can be used as a feature selection technique by calculating the information gain of each variable with respect to the target variable.

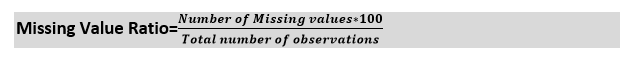
**Chi-square Test:** Chi-square test is a technique to determine the relationship between the categorical variables. The chi-square value is calculated between each feature and the target variable, and the desired number of features with the best chi-square value is selected.

**Fisher's Score:**

Fisher's score is one of the popular supervised technique of features selection. It returns the rank of the variable on the fisher's criteria in descending order. Then we can select the variables with a large fisher's score.

**Missing Value Ratio:**

The value of the missing value ratio can be used for evaluating the feature set against the threshold value. The formula for obtaining the missing value ratio is the number of missing values in each column divided by the total number of observations. The variable is having more than the threshold value can be dropped.



**4.**

**i. Describe the overall feature selection process.**

**Ans:** While developing the ML models, only few features/variables in the dataset are useful for building the model, and rest features are either redundant or irrelevant. which may negatively impact the model performance and increase the model complexity.

For example, Suppose we want to create a model that automatically decides which car should be crushed for a spare part, and to do this, we have a dataset. This dataset contains a Model of the car, Year, Owner's name, Miles. So, in this dataset, the name of the owner does not contribute to the model performance as it does not decide if the car should be crushed or not, so we can remove this column and select the rest of the features(column) for the model building.

For selection the raw data is preprocessed using feature selection techniques as per the domain knowledge and the output to be predicted.

**ii. Explain the key underlying principle of feature extraction using an example. What are the most widely used function extraction algorithms?**

Ans: Feature extraction refers to the process of transforming raw data into numerical features that can be processed while preserving the information in the original data set. It yields better results than applying machine learning directly to the raw data.

Feature extraction is a part of the dimensionality reduction process, in which, an initial set of the raw data is divided and reduced to more manageable groups. So when you want to process it will be easier. The most important characteristic of these large data sets is that they have a large number of variables. These variables require a lot of computing resources to process. So Feature extraction helps to get the best feature from those big data sets by selecting and combining variables into features, thus, effectively reducing the amount of data. These features are easy to process, but still able to describe the actual data set with accuracy and originality

most widely used function extraction algorithms

* Independent component analysis
* Isomap
* Kernel PCA
* Latent semantic analysis
* Partial least squares
* Principal component analysis
* Multifactor dimensionality reduction
* Nonlinear dimensionality reduction
* Semidefinite embedding[Text classification](https://monkeylearn.com/text-classification) is a [machine learning](https://monkeylearn.com/machine-learning/) technique that automatically assigns tags or categories to text. Using [natural language processing (NLP)](https://monkeylearn.com/natural-language-processing/), text classifiers can analyze and sort text by sentiment, topic, and customer intent – faster and more accurately than humans.
* Autoencoder

**5. Describe the feature engineering process in the sense of a text categorization issue.**

Ans: Feature engineering in text categorization is understanding the context of the text. Text categorization is a part of NLP.

Text classification is a machine learning technique that automatically assigns tags or categories to text. Using natural language processing (NLP), text classifiers can analyze and sort text by sentiment, topic, and customer intent – faster and more accurately than humans.

To begin training a classifier with machine learning, you need to transform text into something a machine can understand. This is often carried out using a bag of words, where a vector represents the frequency of a word within a predefined list of words.

Once data is vectorized, the text classifier model is fed training data that consists of feature vectors for each text sample and tag. With enough training samples, the model will be able to make accurate predictions.

**6. What makes cosine similarity a good metric for text categorization? A document-term matrix has two rows with values of (2, 3, 2, 0, 2, 3, 3, 0, 1) and (2, 1, 0, 0, 3, 2, 1, 3, 1). Find the resemblance in cosine.**

Ans: Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction.

*Cosine similarity is one of the metric to measure the text-similarity between two documents irrespective of their size in Natural language Processing*

# Vector representation of document

doc\_1\_vec = [**2, 3, 2, 0, 2, 3, 3, 0, 1]**

doc\_2\_vec = [**2, 1, 0, 0, 3, 2, 1, 3, 1]**

**A.B = ∑ni=1 AiBi**

= (2\*2)+(3\*1)+(2\*0)+(0\*0)+(2\*3)+(3\*2)+(3\*1)+(0\*3)+(1\*1)

=24

√**∑ni=1 A2i =** √4+6+2+0+4+9+9+9+1

=√44 = 6.63

√**∑ni=1 B2i =** √4+2+0+0+9+4+1+9+1

=√30 = 5.47

Cosine similarity = cosθ = =24/(6.63 \*5.47) = 0.66

The Cosine Similarity is a better metric than Euclidean distance because if the two text document far apart by Euclidean distance, there are still chances that they are close to each other in terms of their context.

**7.**

**i. What is the formula for calculating Hamming distance? Between 10001011 and 11001111, calculate the Hamming gap.**

**Ans:** Hamming distance is a metric for comparing two binary data strings. While comparing two binary strings of equal length, Hamming distance is the number of bit positions in which the two bits are different.

In order to calculate the Hamming distance between two strings, and , we perform their XOR operation, (a⊕ b), and then count the total number of 1s in the resultant string

Hamming distance between 10001011 and 11001111, = 2 since the bit varies in two position

i**i. Compare the Jaccard index and similarity matching coefficient of two features with values (1, 1, 0, 0, 1, 0, 1, 1) and (1, 1, 0, 0, 0, 1, 1, 1), respectively (1, 0, 0, 1, 1, 0, 0, 1).**

**Ans:** Jaccard Similarity for Two Sets

This can be written in set notation using intersection ( A ∩ B ) and unions ( A ∪ B ) of two sets: J ( A , B ) = | A ∩ B | | A ∪ B | where | A ∩ B | gives the number of members shared between both sets and | A ∪ B | gives the total number of members in both sets (shared and un-shared).

**J(A, B) = |A** ∩ **B| / |A ∪ B|**

**=** |1,1,1,1,0,0,0|/|1,0| = 7/2 = 3.5

**8. State what is meant by "high-dimensional data set"? Could you offer a few real-life examples? What are the difficulties in using machine learning techniques on a data set with many dimensions? What can be done about it?**

Ans: High-dimensional data are defined as data in which the number of features (variables observed), p, are close to or larger than the number of observations (or data points), n.

Dimensionality in statistics refers to how many attributes a dataset has.

For example, healthcare data is notorious for having vast amounts of variables (e.g. blood pressure, weight, cholesterol level). In an ideal world, this data could be represented in a spreadsheet, with one column representing each dimension. In practice, this is difficult to do, in part because many variables are interrelated (like weight and blood pressure).

As a simple example, let’s say you are using a model to predict the location of a large bacteria in a 25cm2 petri dish. The model might be fairly accurate at pinning the particle down to the nearest square cm. However, let’s say you add just one more dimension: Instead of a 2D petri dish you use a 3D beaker . The predictive space increases exponentially, from 25 cm2 to 125 cm3. When you add more dimensions, it makes sense that **the computational burden also increases**. It wouldn’t be impossible to pinpoint where bacteria might be in a 3D model. However, it’s a more challenging task.

difficulty in using ML techniques on dataset with many dimension:

1. Becomes more difficult to predict and becomes unmanageable
2. Compelxity of model increases
3. requires more computational power

To solve the issue: Feature engineering techniques need to be applied.

1. Feature reduction
2. Feature extraction
3. Feature selection

**9. Make a few quick notes on:**

**1.PCA is an acronym for Personal Computer Analysis.**

PCA in data science refers to Principal Component Analysis. Used for dimension (feature) reduction in large datasets to be used for machine learning, increasing the interpretability of data while preserving the maximum amount of information, and enabling visualization of multidimensional data.

This is accomplished by linearly transforming the data into a new coordinate system where (most of) the variation in the data can be described with fewer dimensions than the initial data

The principal components of a collection of points in a real coordinate space are a sequence of {\displaystyle p}p unit vectors, where the {\displaystyle i}i-th vector is the direction of a line that best fits the data while being orthogonal to the first {\displaystyle i-1}i-1 vectors. Here, a best-fitting line is defined as one that minimizes the average squared perpendicular distance from the points to the line. These directions constitute an orthonormal basis in which different individual dimensions of the data are linearly uncorrelated. Principal component analysis is the process of computing the principal components and using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest.

In data analysis, the first principal component of a set of {\displaystyle p}p variables, presumed to be jointly normally distributed, is the derived variable formed as a linear combination of the original variables that explains the most variance. The second principal component explains the most variance in what is left once the effect of the first component is removed, and we may proceed through {\displaystyle p}p iterations until all the variance is explained. PCA is most commonly used when many of the variables are highly correlated with each other and it is desirable to reduce their number to an independent set.PCA is used in exploratory data analysis and for making predictive models. It is commonly used for dimensionality reduction by projecting each data point onto only the first few principal components to obtain lower-dimensional data while preserving as much of the data's variation as possible. The first principal component can equivalently be defined as a direction that maximizes the variance of the projected data. The {\displaystyle i}i-th principal component can be taken as a direction orthogonal to the first {\displaystyle i-1}i-1 principal components that maximizes the variance of the projected data.

**2. Use of vectors**

create one vector per feature, each containing all observations. Storing data in vectors allows you to leverage linear algebra tools. Note that, even if you can't visualize vectors with a large number of components, you can still apply the same operations on them

In Data Science, vectors are used to represent numeric characteristics, called features, of an object in a mathematical and easily analyzable way. Vectors are essential for many different areas of machine learning and pattern processing.

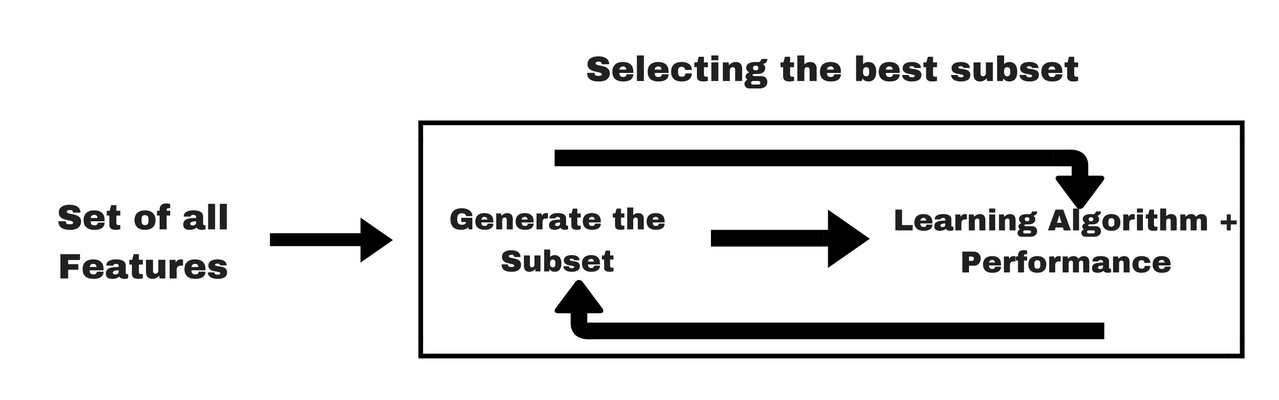
**3. Embedded technique**

Embedded methods combine the qualities’ of filter and wrapper methods. It’s implemented by algorithms that have their own built-in feature selection methods.

Some of the most popular examples of these methods are LASSO and RIDGE regression which have inbuilt penalization functions to reduce overfitting.

Lasso regression performs L1 regularization which adds penalty equivalent to absolute value of the magnitude of coefficients.

Ridge regression performs L2 regularization which adds penalty equivalent to square of the magnitude of coefficients



**10. Make a comparison between:**

**1. Sequential backward exclusion vs. sequential forward selection**

Forward Selection: Forward selection is an iterative method in which we start with having no feature in the model. In each iteration, we keep adding the feature which best improves our model till an addition of a new variable does not improve the performance of the model.

Backward Elimination (Sequential backward exclusion): In backward elimination, we start with all the features and remove the least significant feature at each iteration which improves the performance of the model. We repeat this until no improvement is observed on removal of features.

Recursive Feature elimination: It is a greedy optimization algorithm which aims to find the best performing feature subset. It repeatedly creates models and keeps aside the best or the worst performing feature at each iteration. It constructs the next model with the left features until all the features are exhausted. It then ranks the features based on the order of their elimination.

One of the best ways for implementing feature selection with wrapper methods is to use Boruta package that finds the importance of a feature by creating shadow features.

It works in the following steps:

Firstly, it adds randomness to the given data set by creating shuffled copies of all features (which are called shadow features).

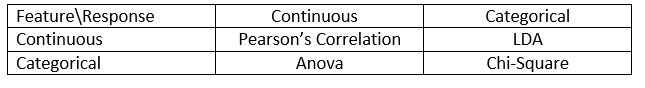
Then, it trains a random forest classifier on the extended data set and applies a feature importance measure (the default is Mean Decrease Accuracy) to evaluate the importance of each feature where higher means more important.

At every iteration, it checks whether a real feature has a higher importance than the best of its shadow features (i.e. whether the feature has a higher Z-score than the maximum Z-score of its shadow features) and constantly removes features which are deemed highly unimportant.

Finally, the algorithm stops either when all features get confirmed or rejected or it reaches a specified limit of random forest runs.

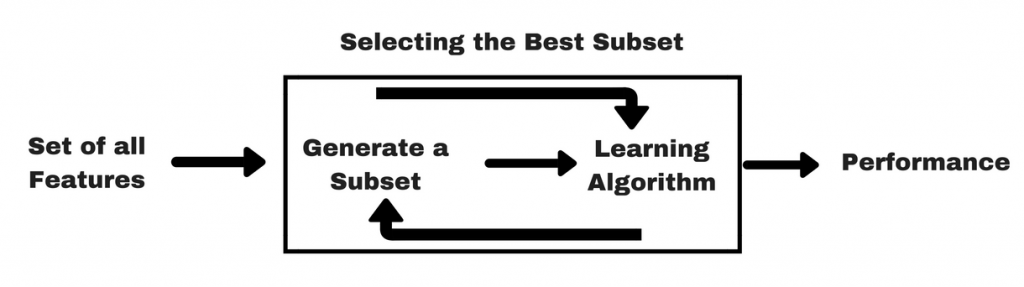
**2. Function selection methods: filter vs. wrapper**

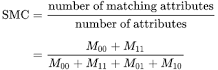
1. Filter:The selection of features is independent of any machine learning algorithms. Instead, features are selected on the basis of their scores in various statistical tests for their correlation with the outcome variable. The correlation is a subjective term here. For basic guidance, you can refer to the following table for defining correlation co-efficients





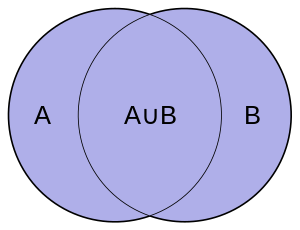
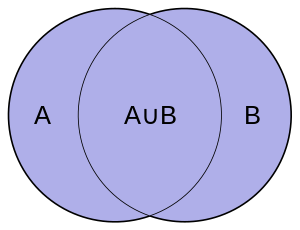
1. Wrapper: In wrapper methods, we try to use a subset of features and train a model using them. Based on the inferences that we draw from the previous model, we decide to add or remove features from your subset. The problem is essentially reduced to a search problem. These methods are usually computationally very expensive.



**3. SMC vs. Jaccard coefficient**

The SMC counts both mutual presences (when an attribute is present in both sets) and mutual absence (when an attribute is absent in both sets) as matches and compares it to the total number of attributes in the universe.

The Jaccard index only counts mutual presence as matches and compares it to the number of attributes that have been chosen by at least one of the two sets



In market basket analysis, for example, the basket of two consumers who we wish to compare might only contain a small fraction of all the available products in the store, so the SMC will usually return very high values of similarities even when the baskets bear very little resemblance, thus making the Jaccard index a more appropriate measure of similarity in that context. For example, consider a supermarket with 1000 products and two customers. The basket of the first customer contains salt and pepper and the basket of the second contains salt and sugar. In this scenario, the similarity between the two baskets as measured by the Jaccard index would be 1/3, but the similarity becomes 0.998 using the SMC.

In other contexts, where 0 and 1 carry equivalent information (symmetry), the SMC is a better measure of similarity. For example, vectors of demographic variables stored in dummy variables, such as binary gender, would be better compared with the SMC than with the Jaccard index since the impact of gender on similarity should be equal, independently of whether male is defined as a 0 and female as a 1 or the other way around. However, when we have symmetric dummy variables, one could replicate the behaviour of the SMC by splitting the dummies into two binary attributes (in this case, male and female), thus transforming them into asymmetric attributes, allowing the use of the Jaccard index without introducing any bias. By using this trick, the Jaccard index can be considered as making the SMC a fully redundant metric. The SMC remains, however, more computationally efficient in the case of symmetric dummy variables since it does not require adding extra dimensions.