Feature engineering:

It is the process of using domain knowledge of the data to create features that make machine learning algorithms work

I will be listing out a total of 15 features that we can use

1. Number of Characters

def count\_chars(text):

return len(text)

2. Number of words

def count\_words(text):

return len(text.split())

3. Number of capital characters

def count\_capital\_chars(text):

count=0

for i in text:

if i.isupper():

count+=1

return count

4. Number of capital words

def count\_capital\_words(text):

return sum(map(str.isupper,text.split()))

5. Count the number of punctuations

def count\_punctuations(text):

punctuations='!"#$%&'()\*+,-./:;<=>?@[\]^\_`{|}~'

d=dict()

for i in punctuations:

d[str(i)+' count']=text.count(i)

return d

6. Number of words in quotes

def count\_words\_in\_quotes(text):

x = re.findall("'.'|"."", text)

count=0

if x is None:

return 0

else:

for i in x:

t=i[1:-1]

count+=count\_words(t)

return count

7. Number of sentences

def count\_sent(text):

return len(nltk.sent\_tokenize(text))

8. Count the number of unique words

def count\_unique\_words(text):

return len(set(text.split()))

10. Count of mentions

def count\_mentions(text):

x = re.findall(r'(@w[A-Za-z0-9]\*)

', text)

return len(x)

11. Count of stopwords

def count\_stopwords(text):

stop\_words = set(stopwords.words('english'))

word\_tokens = word\_tokenize(text)

stopwords\_x = [w for w in word\_tokens if w in stop\_words]

return len(stopwords\_x)

12. Calculating average word length

df['avg\_wordlength'] = df['char\_count']/df['word\_count']

13. Calculating average sentence length

df['avg\_sentlength'] = df['word\_count']/df['sent\_count']

Concept Learning:

In concept learning, we aim to use data to teach a machine to solve a binary classification problem.

That is, to classify a data-point as either belonging to or not belonging to a particular concept or idea.

Features are just columns/attributes derived from our data, and so choosing features for concept learning is, in essence,

choosing a data-viewpoint for the learning problem.

Think aboutTrying to tell whether or not an animal is a dog by only looking at its height

On the pytheon the other hand, if we choose to include too many features in our learning model, the noise

and increased complexity could make the learning process significantly harder.

The initial set of raw features can be redundant and too large to be managed. Therefore, a preliminary step in many applications of machine learning

and pattern recognition consists of selecting a subset of features, or constructing a new and reduced set of features to facilitate learning,

and to improve generalization and interpretability

Feature Extraction:

Feature extraction is a process of dimensionality reduction by which an initial set of raw data is reduced to more manageable groups for processing.

A characteristic of these large data sets is a large number of variables that require a lot of computing resources to process.

Feature extraction is the name for methods that select and /or combine variables into features, effectively reducing the amount of

data that must be processed, while still accurately and completely

describing the original data set.

Practical Uses of Feature Extraction:

Autoencoders:

Bag-of-Words

– A technique for natural language processing that extracts the words (features) used in a sentence, document, website, etc

Image Processing – Algorithms are used to detect features such as shaped, edges, or motion in a digital image or video.

Using Regularization could certainly help reduce the risk of overfitting, but using instead Feature Extraction techniques can also lead to other types of advantages such as:

Accuracy improvements.

Overfitting risk reduction.

Speed up in training.

Improved Data Visualization.

Increase in explainability of our model.

These new reduced set of features should then be able to summarize most of the information contained in the original set of features. In this way, a summarised version of the

original features can be created from a combination of the original set.

Another commonly used technique to reduce the number of feature in a dataset is Feature Selection

The difference between Feature Selection and Feature Extraction is that feature selection aims instead to rank the importance of the existing features in the dataset

and discard less important ones (no new features are created).

PCA is one of the most used linear dimensionality reduction technique.

The difference between Feature Selection and Feature Extraction is that feature selection aims instead to rank the importance

of the existing features in the dataset and discard less important ones (no new features are created)

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

A. Filter methods

B. Wrapper methods

C. Embedded methods

D. Hybrid methods

A. Filter methods

Filter methods are generally used as a preprocessing step. 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.

Pearson’s Correlation: It is used as a measure for quantifying linear dependence between two continuous variables X and Y.

Its value varies from -1 to +1. Pearson’s correlation is given as:fs2

LDA: Linear discriminant analysis is used to find a linear combination of features that characterizes or separates two or more classes (or levels) of a categorical variable.

ANOVA: ANOVA stands for Analysis of variance. It is similar to LDA except for the fact that it is operated using one or more categorical independent features and one continuous dependent feature. It provides a statistical test of whether the means of several groups are equal or not.

Chi-Square: It is a is a statistical test applied to the groups of categorical features to evaluate the likelihood of correlation or association between them using their frequency distribution.

In statistics, multicollinearity (also collinearity) is a phenomenon in which one predictor variable in

a multiple regression model can be linearly predicted from the others with a substantial degree of accuracy.

Multicollinearity occurs when two or more independent variables are highly correlated with one another in a regression model.

For example, height and weight, household income and water consumption, mileage and price of a car, study time and leisure time, etc.

Let me take a simple example from our everyday life to explain this. Colin loves watching television while munching on chips.

The more television he watches, the more chips he eats and the happier he gets!

Now, if we could quantify happiness and measure Colin’s happiness while he’s busy doing his favorite activity,

which do you think would have a greater impact on his happiness? Having chips or watching television?

That’s difficult to determine because the moment we try to measure Colin’s happiness from eating chips,

he starts watching television. And the moment we try to measure his happiness from watching television,

he starts eating chips.

Multicollinearity can be a problem in a regression model because we would not be able to distinguish between the individual

effects of the independent variables on the dependent variable.

For example, let’s assume that in the following linear equation:

Y = W0+W1\*X1+W2\*X2

Coefficient W1 is the increase in Y for a unit increase in X1 while keeping X2 constant.

But since X1 and X2 are highly correlated, changes in X1 would also cause changes in X2 and we would not be able to see their individual effect on Y.

Eating chips and watching television are highly correlated in the case of Colin and we cannot individually

determine the impact of the individual activities on his happiness. This is the multicollinearity problem!

This makes the effects of X1 on Y difficult to distinguish from the effects of X2 on Y. ”

So why should you worry about multicollinearity in the machine learning context? Let’s answer that question next.

Multicollinearity may not affect the accuracy of the model as much.

But we might lose reliability in determining the effects of individual features in your model – and that can be a problem when it comes to interpretability.

Inaccurate use of dummy variables can also cause a multicollinearity problem. This is called the Dummy variable trap:

Wrapper Methods: Definition

Wrapper methods work by evaluating a subset of features using a machine learning algorithm that employs a search strategy

to look through the space of possible feature subsets, evaluating each subset based on the quality of the performance of a given algorithm.

These methods are called greedy algorithms because they aim to find the best possible combination of features that

result in the best performant model— which will be computationally expensive, and often impractical in the case of exhaustive search.

Practically any combination of a search strategy and a machine learning algorithm can be used as a wrapper.

Wrapper Methods: Advantages

Given the issues we encountered using filter methods, wrapper methods present two main advantages that deal with those issues:

They detect the interaction between variables

They find the optimal feature subset for the desired machine learning algorithm

The wrapper methods usually result in better predictive accuracy than filter methods.

Wrapper Methods: Process

Wrapper methods work in the following way, generally speaking:

Search for a subset of features: Using a search method (described next), we select a subset of features from the available ones.

Build a machine learning model: In this step, a chosen ML algorithm is trained on the previously-selected subset of features.

Evaluate model performance: And finally, we evaluate the newly-trained ML model with a chosen metric.

Repeat: The whole process starts again with a new subset of features, a new ML model trained, and so on.

We stop until the desired condition is met, and then we choose the best subset with the best result in the evaluation phase.

At some point in time, we need to stop searching for a subset of features. To do this, we have to put in place some

pre-set criteria (often somewhat arbitrary), around when this searching should stop. These criteria need to be defined by the machine learning engineer. Here are a couple of examples of these criteria:

Model performance decreases.

Model performance increases.

A predefined number of features is reached.

The pre-set criteria, for example, can be metrics like ROC-AUC for classification or RMSE for regression.

Search methods

Forward Feature Selection: This method starts with no feature and adds one at a time.

Backward Feature Elimination: This method starts with all features present and removes one feature at the time.

Exhaustive Feature Selection: This method tries all possible feature combinations.

Bidirectional Search: And this last one does both forward and backward feature selection simultaneously in order to get one unique solution.

Embedded Methods: Definition

Embedded methods complete the feature selection process within the construction of the machine learning algorithm itself.

In other words, they perform feature selection during the model training, which is why we call them embedded methods.

A learning algorithm takes advantage of its own variable selection process and performs feature selection and classification/regression at the same time.

Embedded Methods: Advantages

The embedded method solves both issues we encountered with the filter and wrapper methods by combining their advantages. Here’s how:

They take into consideration the interaction of features like wrapper methods do.

They are faster like filter methods.

They are more accurate than filter methods.

They find the feature subset for the algorithm being trained.

They are much less prone to overfitting.

Embedded Methods: Process

Any and all embedded methods work as follows:

First, these methods train a machine learning model.

They then derive feature importance from this model, which is a measure of how much is feature important when making a prediction.

Finally, they remove non-important features using the derived feature importance.

Hybrid methods:

Hybrid Methods: Definition

The definition here is quite simple—rather than using a single approach to select feature subsets as the previous methods do,

hybrid methods combine the different approaches to get the best possible feature subset.

The way to combine these approaches is up to the engineer, given that you have a lot of methods in your toolbox.

You can, for example, start by performing filter methods by eliminating constant, quasi-constant and duplicated features.

Then, in the second step, you could use wrapper methods to select the best feature subset from the previous step.

This is just one simple, high-level approach.

What Is Information Gain?

Information Gain, or IG for short, measures the reduction in entropy or surprise by splitting a dataset according to a given value of a random variable.

A larger information gain suggests a lower entropy group or groups of samples, and hence less surprise.

You might recall that information quantifies how surprising an event is in bits.

Lower probability events have more information, higher probability events have less information.

Entropy quantifies how much information there is in a random variable, or

more specifically its probability distribution. A skewed distribution has a low entropy,

whereas a distribution where events have equal probability has a larger entropy.

In information theory, we like to describe the “surprise” of an event.

Low probability events are more surprising therefore have a larger amount of information.

Whereas probability distributions where the events are equally likely are more surprising and have larger entropy.

The information gain is calculated for each variable in the dataset. The variable that has the largest information gain is selected to split the dataset.

Generally, a larger gain indicates a smaller entropy or less surprise.

Note that minimizing the entropy is equivalent to maximizing the information gain

the process of selecting a new attribute and partitioning the training examples is now repeated

for each non terminal descendant node, this time using only the training examples associated with that node.

Attributes that have been incorporated higher in the tree are

excluded, so that any given attribute can appear at most once along any path through the tree.

mutual information and Chi score:

Mutual information is straightforward when considering the distribution of two discrete (categorical or ordinal) variables, such as categorical

input and categorical output data. Nevertheless, it can be adapted for use with numerical input and categorical output.

ANOVA f-test Feature Selection

ANOVA is an acronym for “analysis of variance” and is a parametric statistical hypothesis test for determining

whether the means from two or more samples of data (often three or more) come from the same distribution or not.

An F-statistic, or F-test, is a class of statistical tests that calculate the ratio between variances values,

such as the variance from two different samples or the explained and unexplained variance by a statistical test,

like ANOVA. The ANOVA method is a type of F-statistic referred to here as an ANOVA f-test.

Importantly, ANOVA is used when one variable is numeric and one is categorical, such as numerical input

variables and a classification target variable in a classification task.

The results of this test can be used for feature selection where those features that are independent

of the target variable can be removed from the dataset.

chi-square measures the statistical significance of the differences between the child nodes and their parent nodes.

It is measured as the sum of squared standardized differences between observed and expected frequencies

of target variable for each node and is calculated using this formula-

Chi-square measures the statistical significance of the differences between the child nodes and their parent nodes.

It is measured as the sum of squared standardized differences between observed and expected frequencies of

target variable for each node and is calculated using this formula-

We’ve seen this before. There is a total of 20 students and out of those 10 play cricket and 10 do not.

So, of course, the percent of students who do play cricket will be 50%. Now if we consider the “Above average”

node here, there are 14 students in it, as the percentage of students who play cricket is 50% in the parent

node as we discussed, the expected number of students who play cricket will of course be 7 and if you look

at the actual value it is 8. So now we have both the values expected values and actual values.

The expected was 7 and the actual turns out to be 8, so we can infer that the expected value is calculated

based on the distribution of the parent node of the same class. Similarly, the expected number of students

who do not play cricket will be 7. I want you to intuitively think about this because remember the percentage

of students who do not play cricket in the parent node is 50% as well and then here the actual value turns out to be 6.

Similarly for the below-average node expected to play and not play will be 3. Whereas the actual values are- 2

students play cricket and 4 do not.

Now we can calculate the chi-square using the formula mentioned above for each child node.

Can you guess what will be the chi-square value if the actual and expected values are the same?

It’s actually pretty simple, it will be zero because both the actual and expected are the same and the difference will be zero.

Now if both values are the same we can generate an inference that the distribution of the child

node is the same as there is the parent node and hence we are not improving the purity of the nodes.

On the other hand, if the chi-square value is high it means that the distribution of child nodes is

changing with respect to the parent node and we are going in a direction to

achieve more pure nodes hence we can say that:

Higher the chi-square value more will be the purity of the nodes after a split.

Properties of chi-square

Let’s look at some of the properties of chi-square before understanding how it actually works-

Chi-square just like Gini impurity works only with categorical variables so we cannot use it for continuous targets.

The higher the value of chi-square more the sub-nodes are different from the parent node and hence the homogeneity is more.

These are some of the properties of chi-square and you must consider these properties before choosing the right algorithm for deciding the split.

Properties of the Chi-Square

Chi-square is non-negative. Is the ratio of two non-negative values, therefore must be non-negative itself.

Chi-square is non-symmetric.

There are many different chi-square distributions, one for each degree of freedom.

The degrees of freedom when working with a single population variance is n-1.

Chi-Square Probabilities

Since the chi-square distribution isn't symmetric, the method for looking up left-tail values

is different from the method for looking up right tail values.

Area to the right - just use the area given.

Area to the left - the table requires the area to the right, so subtract the given area from one and look this area up in the table.

Area in both tails - divide the area by two. Look up this area for the right critical value and one minus this area for the left critical value.

Properties of Chi-Square Distribution

The chi-square distribution is a continuous probability distribution with the values ranging from 0 to ∞ (infinity) in the positive

direction. The χ2 can never assume negative values.

The shape of the chi-square distribution depends on the number of degrees of freedom ‘ν’.

When ‘ν’ is small, the shape of the curve tends to be skewed to the right, and as the ‘ν’ gets larger, the shape becomes more

symmetrical and can be approximated by the normal distribution.

The mean of the chi-square distribution is equal to the degrees of freedom, i.e. E(χ2) = ‘ν’.

While the variance is twice the degrees of freedom, Viz. n(χ2) = 2ν.

The χ2 distribution approaches the normal distribution as ν gets larger with mean ν and standard

deviation as √2χ2. It has been determined that quantity √2χ2 gives a better approximation to normality than

the χ2 itself if the values are about 30 or more. Thus, the mean and standard deviation of the distribution

of √2χ2 is equal to √2ν-1 and one respectively.

The sum of independent χ2 is itself a χ2 variate. Suppose, χ12 is a χ2 variate with degrees of freedom ν1

and χ22 is another χ2 variate with degrees of freedom ν2, then their sum χ12 + χ22 will be equal to χ2 variate with

ν1+ ν2 degrees of freedom. This property is called as the additive property of Chi-square.

Thus, χ2 distribution depends on the degrees of distribution as its shape changes with the change in the ‘ν’

, and as ‘ν’ becomes greater, χ2 gets approximated by the normal distribution.

Degree of Freedom:

The Degrees of Freedom refers to the number of values involved in the calculations that have the freedom to vary.

In other words, the degrees of freedom, in general, can be defined as the total number

of observations minus the number of independent constraints imposed on the observations.

Let’s consider a scenario where we need to determine the relationship between the independent category feature (predictor)

and dependent category feature(response). In feature selection,

we aim to select the features which are highly dependent on the response.

When two features are independent, the observed count is close to the expected count, thus we will have

smaller Chi-Square value. So high Chi-Square value indicates that the hypothesis of independence is incorrect. In simple words, higher the Chi-Square value

the feature is more dependent on the response and it can be selected for model training.

What is the AUC-ROC curve?

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems.

It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the ‘signal’ from the ‘noise’. The Area Under the Curve (AUC) is the measure of the ability of a classifier to

distinguish between classes and is used as a summary of the ROC curve.

When AUC = 1, then the classifier is able to perfectly distinguish between all the Positive and the Negative

class points correctly. If, however, the AUC had been 0, then the classifier would be predicting all Negatives as Positives,

and all Positives as Negatives.

When 0.5<AUC<1, there is a high chance that the classifier will be able to distinguish the positive class

values from the negative class values. This is so because the classifier is able to detect more numbers

of True positives and True negatives than False negatives and False positives.

When AUC=0.5, then the classifier is not able to distinguish between Positive and Negative class points. Meaning either the classifier is

predicting random class or constant class for all the data points.

In a ROC curve, a higher X-axis value indicates a higher number of False positives than True negatives.

While a higher Y-axis value indicates a higher number of True positives than False negatives. So, the choice of the

threshold depends on the ability to balance between False positives and False negatives.

Let’s dig a bit deeper and understand how our ROC curve would look like for different threshold values

and how the specificity and sensitivity would vary.

**ROC curve**

An **ROC curve** (**receiver operating characteristic curve**) is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters:

* True Positive Rate
* False Positive Rate

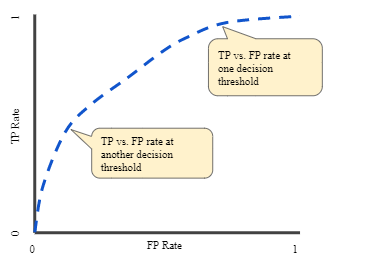
**True Positive Rate** (**TPR**) is a synonym for recall and is therefore defined as follows:

TPR=TP/TP+FN

**False Positive Rate** (**FPR**) is defined as follows:

FPR=FP/FP+TN

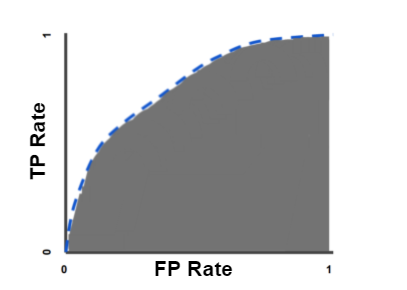
An ROC curve plots TPR vs. FPR at different classification thresholds. Lowering the classification threshold classifies more items as positive, thus increasing both False Positives and True Positives. The following figure shows a typical ROC curve.



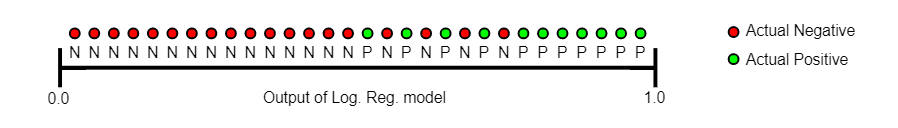
To compute the points in an ROC curve, we could evaluate a logistic regression model many times with different classification thresholds, but this would be inefficient. Fortunately, there's an efficient, sorting-based algorithm that can provide this information for us, called AUC.

AUC:

**AUC** stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve (think integral calculus) from (0,0) to (1,1).



AUC provides an aggregate measure of performance across all possible classification thresholds. One way of interpreting AUC is as the probability that the model ranks a random positive example more highly than a random negative example. For example, given the following examples, which are arranged from left to right in ascending order of logistic regression predictions:



AUC represents the probability that a random positive (green) example is positioned to the right of a random negative (red) example.

AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.

AUC is desirable for the following two reasons:

* AUC is **scale-invariant**. It measures how well predictions are ranked, rather than their absolute values.
* AUC is **classification-threshold-invariant**. It measures the quality of the model's predictions irrespective of what classification threshold is chosen.

However, both these reasons come with caveats, which may limit the usefulness of AUC in certain use cases:

* **Scale invariance is not always desirable.** For example, sometimes we really do need well calibrated probability outputs, and AUC won’t tell us about that.
* **Classification-threshold invariance is not always desirable.** In cases where there are wide disparities in the cost of false negatives vs. false positives, it may be critical to minimize one type of classification error. For example, when doing email spam detection, you likely want to prioritize minimizing false positives (even if that results in a significant increase of false negatives). AUC isn't a useful metric for this type of optimization.

Feature Selection :

## Introduction

Feature selection has become increasingly important for data analysis, machine learning, and data mining. Especially for high-dimensional data sets, it is necessary to filter out the irrelevant and redundant features by choosing a suitable subset of relevant features in order to avoid over-fitting and tackle the curse of dimensionality. With respect to data sets from the bioinformatics domain, feature selection often allows identifying the features that are important for biological processes of interest.

When fitting a statistical model for such high-dimensional data sets, one needs to decide first which feature selection method to use. As many of them exist, this is not an easy decision. Regarding the comparison of different methods, benchmark studies have gained increasing attention in the machine learning community ([Fernández-Delgado et al., 2014](https://www.sciencedirect.com/science/article/pii/S016794731930194X" \l "b18)).

In this paper, we benchmark state-of-the-art feature selection techniques on high-dimensional data sets. We compare 22 filter methods from different toolboxes on 16 high-dimensional classification data sets from various domains. We investigate which methods select the features of a data set in a similar order. Additionally, we search for optimal methods with respect to predictive accuracy and run time. The results of our analyses show three groups of similar filter methods as well as several filter methods that are not very similar to any other filter method. Some of the filter methods seem to work better than others, but which filter methods perform best depends on the data set. There is no one-fits-all solution, but some recommendations can be made on the choice of filter methods.

In the past decades, many feature selection methods have been proposed. The methods can be categorized into three classes ([Guyon and Elisseeff, 2003](https://www.sciencedirect.com/science/article/pii/S016794731930194X" \l "b22)): Filter methods rank features by calculating a score for each feature independent of a model. Either the m features with the highest scores or all the features whose scores exceed a threshold τ are selected (with m∈N or τ∈R being pre-specified). For many filter methods, the score calculation can be done in parallel. [Lazar et al. (2012)](https://www.sciencedirect.com/science/article/pii/S016794731930194X" \l "b41) give an extensive overview of existing filter methods.

All filter methods described in this section are applicable for classification data sets with numeric features. Some of the methods also work for categorical features. We present two kinds of filter methods: Most filter methods calculate a score for all features and then select the features with the highest scores. Some filter methods, however, select features iteratively in a greedy forward fashion. For these filters, in each iteration the feature with the maximal score is selected but the scores of different iterations are not comparable. Notation-wise, we consider a data set with n instances of the p features X1,…,Xp and class variable Y. The filter methods in Sections [2.1](https://www.sciencedirect.com/science/article/pii/S016794731930194X" \l "sec2.1)–[2.3](https://www.sciencedirect.com/science/article/pii/S016794731930194X" \l "sec2.3) are univariate, i.e. they do not consider interactions between the p features. Most of the filter methods in Sections [2.4 Random forest importance](https://www.sciencedirect.com/science/article/pii/S016794731930194X" \l "sec2.4), [2.5 Mutual information](https://www.sciencedirect.com/science/article/pii/S016794731930194X" \l "sec2.5) are multivariate.

### Univariate statistical tests

Filter anova.test performs for each feature an analysis of variance where the class variable is explained by the feature. The value of the F statistic is used as the score. The higher the F statistic, the more different are the mean values of the corresponding feature between the classes. For each feature Xk, the filter score is defined as

