NOVEL FEATURE SELECTION METHODS FOR WATER QUALITY DATA

**Team Members:**

1. Shashank D - 181CO248
2. Tarun S Anur - 181CO255

# INTRODUCTION

Feature selection is a process of reducing the number of input variables during the development of a predictive model.

This is mostly done in order to reduce the computational costs and in some cases even increase the performance of the model to some extent. Feature selection is primarily focused on removing non-informative or redundant predictors from the model.

Statistical-based feature selection methods involve evaluating the relationship between each input variable and the target variable using statistics and selecting those input variables that have the strongest relationship with the target variable. These methods can be fast and effective, although the choice of statistical measures depends on the data type of both the input and output variables.

We can divide the feature selection methods broadly into three types, namely, **WRAPPER**, **FILTER** and **INTRINSIC** methods:

**Wrapper feature selection** methods create many models with different subsets of input features and select those features that result in the best performing model according to a given performance metric. These methods are unconcerned with the variable types, although they can be computationally expensive. **RFE (Recursive Feature Elimination)** is a good example of a wrapper feature selection method. Wrapper methods evaluate multiple models using procedures that add and/or remove predictors to find the optimal combination that maximizes model performance.

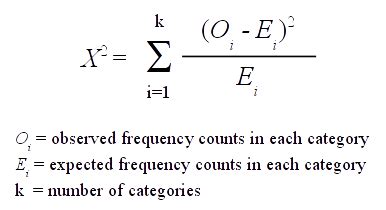
**Filter feature selection** methods use statistical techniques to evaluate the relationship between each input variable and the target variable, and these scores are used as the basis to choose and filter out those input variables that will be used in the model. Filter methods evaluate the relevance of the predictors outside of the predictive models and subsequently model only the predictors that pass some criterion. **Feature Importance** is the most commonly used filter feature selection method.

**Intrinsic feature selection** methods where automatic feature selection is performed automatically during the learning of the model. **Decision Tree classifier** is one the commonly used intrinsic feature selection method.

## DIFFERENT FEATURE SELECTION METHODS USED:

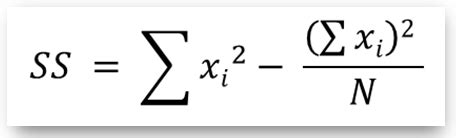
* UNIVARIATE SELECTION USING CHI-SQUARE TEST

This method computes chi-squared stats between each non-negative feature and class. This score can then be used to select the n\_features with the highest values for the test chi-squared statistic from X, which must contain only non-negative features such as Booleans or frequencies relative to the classes. It is a statistical test applied to sets of [categorical data](https://en.wikipedia.org/wiki/Categorical_data) to evaluate how likely it is that any observed difference between the sets arose by chance. The chi-square score is calculated by the below formula:



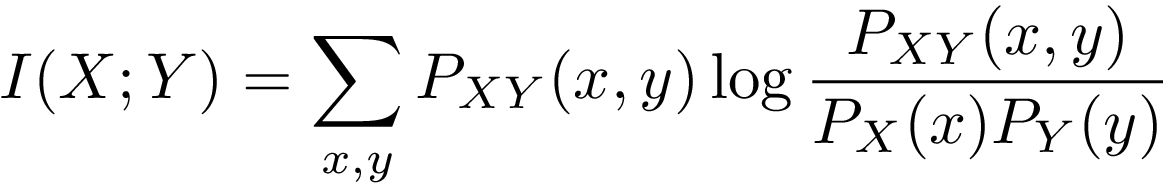
* UNIVARIATE SELECTION USING F\_CLASSIF METHOD (ANOVA TEST)

ANOVA (Analysis of Variance) is another method to evaluate the dependence of two variables. ANOVA assumes a linear relationship between the variables and the target, and also that the variables are normally distributed. ANOVA is a collection of [statistical models](https://en.wikipedia.org/wiki/Statistical_model) and their associated estimation procedures used to analyse the differences among means. ANOVA is based on the [law of total variance](https://en.wikipedia.org/wiki/Law_of_total_variance), where the observed [variance](https://en.wikipedia.org/wiki/Variance) in a particular variable is partitioned into components attributable to different sources of variation.



* UNIVARIATE SELECTION USING MUTUAL\_INFO\_CLASSIF METHOD

This method estimates Mutual Information (MI) for a discrete target variable. It quantifies the "[amount of information](https://en.wikipedia.org/wiki/Information_content)" obtained about one random variable through observing the other random variable. MI between two random variables is a non-negative value, which measures the dependency between the variables. It is equal to zero if and only if two random variables are independent, and higher values mean higher dependency. MI measures how much information the presence/absence of a feature X contributes to making the correct prediction on Y. If X is deterministic of Y, then MI is the entropy of X, which is a notion in information theory that measures or quantifies the amount of information within a variable. True MI can never be negative. In case it turns out negative, it is replaced by 0.



* FEATURE IMPORTANCES USING EXTRA RANDOMIZED TREE CLASSIFIER

Each Decision Tree in the Extra Trees Forest is constructed from the original training sample. Then, at each test node, each tree is provided with a random sample of k features from the feature-set from which each decision tree must select the best feature to split the data based on some mathematical criteria (typically the Gini Index). This random sample of features leads to the creation of multiple de-correlated decision trees. This class implements a meta estimator that fits a number of randomized decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting. Extra-trees differ from classic decision trees in the way they are built. When looking for the best split to separate the samples of a node into two groups, random splits are drawn for each of the max\_features randomly selected features and the best split among those is chosen.

* SEQUENTIAL FEATURES SELECTOR (FORWARD SELECTION)

This Sequential Feature Selector adds (forward selection) features to form a feature subset in a greedy fashion. At each stage, this estimator chooses the best feature to add or remove based on the cross-validation score of an estimator (decision tree classifier). Forward-SFS is a greedy procedure that iteratively finds the best new feature to add to the set of selected features. Concretely, we initially start with zero feature and find the one feature that maximizes a cross-validated score when an estimator is trained on this single feature. Once that first feature is selected, we repeat the procedure by adding a new feature to the set of selected features. The procedure stops when the desired number of selected features is reached.

* SEQUENTIAL FEATURES SELECTOR (BACKWARD SELECTION)

This Sequential Feature Selector removes (backward selection) features to form a feature subset in a greedy fashion. At each stage, this estimator chooses the best feature to remove based on the cross-validation score of an estimator (decision tree classifier). Backward-SBS is a greedy procedure that iteratively finds the best new feature to remove from the set of selected features. Concretely, we initially start with all features and find the one feature that minimizes a cross-validated score when an estimator is trained using all these features. Once that first feature is removed, we repeat the procedure by removing a feature to the set of selected features. This feature can be also described as the least significant feature among the remaining available ones. The procedure stops when the desired number of selected features is reached.

* SEQUENTIAL FEATURES SELECTOR USING KNN ESTIMATOR (FORWARD SELECTION)

This Sequential Feature Selector adds (forward selection) features to form a feature subset in a greedy fashion. At each stage, this estimator chooses the best feature to add or remove based on the cross-validation score of an estimator (K – Nearest Neighbours). Forward-SFS is a greedy procedure that iteratively finds the best new feature to add to the set of selected features. Concretely, we initially start with zero feature and find the one feature that maximizes a cross-validated score when an estimator is trained on this single feature. Once that first feature is selected, we repeat the procedure by adding a new feature to the set of selected features. The procedure stops when the desired number of selected features is reached.

* RECURSIVE FEATURE ELIMINATION WITH DECISION TREE ESTIMATOR

The goal of Recursive Feature Elimination (RFE) is to select features by recursively considering smaller and smaller sets of features. RFE works by searching for a subset of features by starting with all features in the training dataset and successfully removing features until the desired number remains. This is achieved by fitting the given machine learning algorithm used in the core of the model, ranking features by importance, discarding the least important features, and re-fitting the model. This process is repeated until a specified number of features remains. First, the estimator is trained on the initial set of features and the importance of each feature is obtained either through any specific attribute or callable. Then, the least important features are pruned from current set of features. That procedure is recursively repeated on the pruned set until the desired number of features to select is eventually reached.

* CORRELATION CO-EFFICIENTS ANALYSIS

A correlation co-efficient is a [numerical measure](https://en.wikipedia.org/wiki/Numerical_measure) of some type of [correlation](https://en.wikipedia.org/wiki/Correlation_and_dependence), meaning a statistical relationship between two [variables](https://en.wikipedia.org/wiki/Variable_(mathematics)). It is used to find the pairwise correlation of all columns in the dataframe. They all assume values in the range from −1 to +1, where ±1 indicates the strongest possible agreement and 0 the strongest possible disagreement. The [Pearson correlation coefficient](https://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient), is a measure of the strength and direction of the linear relationship between two variables that is defined as the [covariance](https://en.wikipedia.org/wiki/Covariance) of the variables divided by the product of their standard deviations. This is the best-known and most commonly used type of correlation coefficient.

* UNIVARIATE SELECTION USING THE G-TEST