

# USER MANUAL

## **SFS-QSAR-tool** **(Version 2.0.0)**

Department of Biochemistry and Chemistry  
FACULDADE DE CIÊNCIAS DA  
UNIVERSIDADE DO PORTO  
Rua do Campo Alegre, s/n, 4169-007  
Porto, Portugal  
[www.fc.up.pt](http://www.fc.up.pt)

## Overview of the tool

*SFS-QSAR-tool\_v2* is an updated version of the SFS-QSAR-tool available in public domain (<https://github.com/ncordeirfcup/SFS-QSAR-tool>). SFS-QSAR-tool was developed for performing QSAR modelling with the help of sequential forward selection technique. Sequential forward selection is a feature selection technique used in machine learning to select a subset of relevant features for model development. The goal of feature selection is to improve the performance of the model by reducing overfitting, enhancing generalization, and simplifying the model. SFS evaluates all features individually by adding each feature to the current set of selected features and training a model. The process is continued until a predefined number of features is selected, or adding more features does not significantly improve the model's performance.

In contrast to sequential backward selection (SBS), which identifies the most relevant subset of features for a model by sequentially removing the least significant features, it starts with an empty set and adds features one by one. SFS is wrapper method of feature selection and generally provides a better indication than filter methods, but at a higher computational expense.

**Statistical parameters:** A number of different statistical parameters can be used to determine the goodness of fit of the developed models. Parameters like  $Q^2_{\text{LOO}}$  and  $R^2_{\text{Pred}}$  give the best measure of the predictivity. Apart from this, parameters like determination coefficient  $R^2$ , adjusted  $R^2$  ( $R^2_{\text{Adj}}$ ), Fischer statistic ( $F$ -test), mean absolute error (MAE), and the metrics  $r_m^2_{\text{LOO}}$  and  $\Delta r_m^2_{\text{LOO}}$  for the training set, as well as  $r_m^2_{\text{test}}$  and  $\Delta r_m^2_{\text{test}}$  for the test set can be determined for determining the predictive quality of the models. The unique nature of the model may be calculated by determining the  $^cR^2_{\text{p}}$  value through Y-randomization test.

**Applicability domain (AD):** The applicability domain refers to the range of chemical structures, data, or conditions under which a predictive model or methodology is considered reliable and accurate. The major purpose for determining AD of a model is to identify the reliability of the model. AD also helps in avoiding making predictions outside the scope of model's training, which can ultimately lead to incorrect and unreliable predictions. SFS-QSAR-tool measures the AD of a developed model by computing Williams plot which also known as the residual plot or residual-versus-time plot. the Williams plot serves as a diagnostic tool to assess the performance and reliability of QSAR models. This plot aids in identifying outliers in both response and structure by illustrating the influence of each data point through leverage against

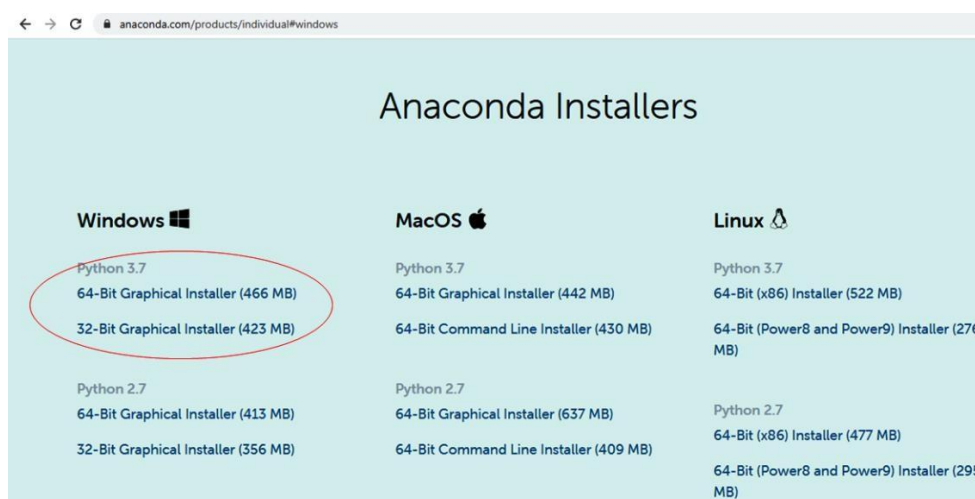
standardized residuals.

**FS-LDA:** The independent descriptors are included in the model stepwise depending on specific statistical parameter. Here, the features are selected and included in the model stepwise by the p-values in an F-statistic. Initially, criteria for forward selection (i.e. p-value to enter) and backward elimination (p-value to remove) are set. The descriptor with the lowest p-value is included first and subsequently other descriptors are included in the model based on the lowest p-value only if the criteria for forward selection is met. However, if the p-value of a descriptor included in the model is found to be greater than ‘p-value to remove’, it is eliminated from the model. In the current work, both p-to enter and p-to remove are fixed as 0.05. The final LDA models were developed and were subsequently validated using *LinearDiscriminantAnalysis* function of Scikit-learn.

**SFS-LDA:** It adds features into an empty set until the performance of the model is not improved either by addition of another feature or maximum number of features is reached. Similar to FS-LDA this technique is also a greedy search algorithm where the best subsets of descriptors are selected stepwise and the model performance is justified by the user specific statistical measure.

### Installation of dependencies of *SFS-QSAR-tool\_v2*:

1. SFS-QSAR-tool\_v2 is a python-based tool that has multiple dependencies. Therefore, the users should install anaconda (<https://www.anaconda.com/products/individual#windows>) with python-3.



2. Additionally, the user needs to install some dependencies (see below). For this, open anaconda and type ‘`pip install -r requirements.txt`’.

```

Anaconda Powershell Prompt
(base) PS C:\Users\Soumya> cd C:\Users\Soumya\Documents\SFS-QSAR-tool_v2-main
(base) PS C:\Users\Soumya\Documents\SFS-QSAR-tool_v2-main> pip install -r requirements.txt
```

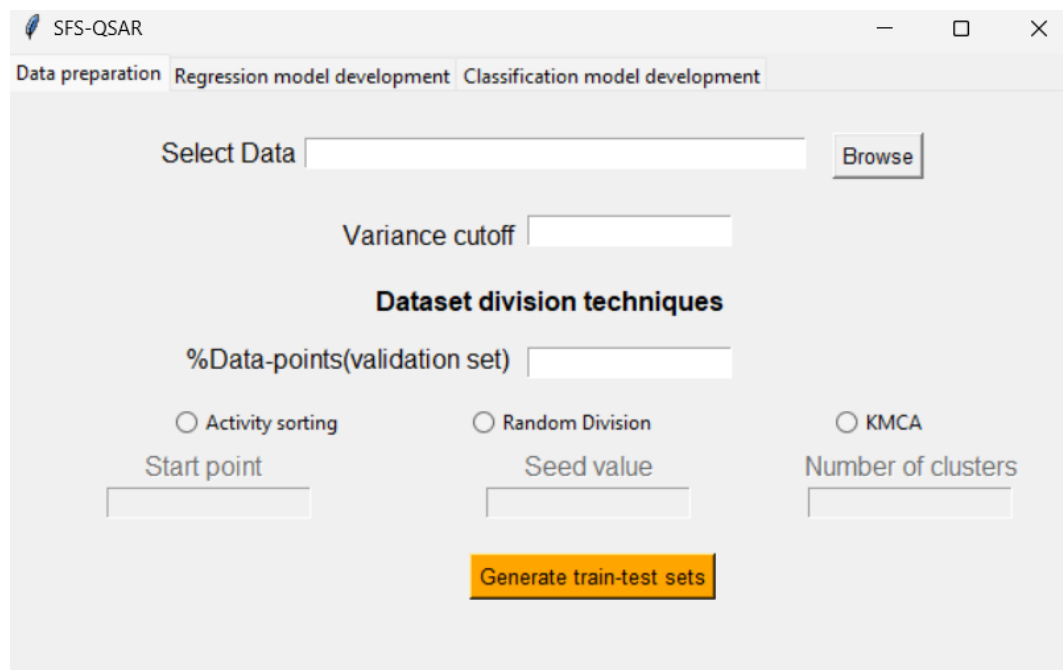
3. The Mlxtend should be installed with the following command in Anaconda:

**conda install -c conda-forge mlxtend**

### Operating SFS-QSAR-tool\_v2

1. To operate the program, first go to the directory containing the tool.
2. To run the program, use the following command in Anaconda:

**python SFS-QSAR4.py**



3. Once run, the program will start with the following graphical interface.
4. As seen in the interface, there are three (03) tabs present namely, **a. Data preparation, b. Regression model development and c. Classification model development.**

#### ***a. Data preparation***

1. The operation starts with Data preparation where the complete dataset (.csv file) containing the names, dependable variables and independent variables is selected. A model complete dataset has been provided for understanding (***data.csv***). The first column of the dataset should act as index for the dataset while the second column typically contains the independent variable. Rest of the columns contain dependent variable which are typically molecular descriptors for performing QSAR analysis.
2. In the next step, data pre-treatment is required to eliminate near constant variables. To carry out, data pre-treatment initially variance cut-off may be kept at 0.001.
3. Once the data pretreatment is completed, the next step involves dataset division. However, before choosing the preferred technique to carry out dataset division, at first mentioning the percentage (%) of datapoints that should be placed in test set (validation set) is important. Typically, for relatively smaller datasets, 20% of total datapoints are placed in test set

(validation set). For dataset division, three options are available.

(i) *Activity sorting*

(ii) *Random Division*

(iii) *KMCA (k-means cluster analysis)*

#### *Activity sorting*

Activity sorting as dataset division technique typically involves categorizing activities into different groups or segments based on specific criteria. In this technique, at first the tool extracts the names of column from the dataset except for the first column. Afterwards, the complete dataset is rearranged according to the activity, which is typically the independent variable, in descending order. Then, the splitting parameter is set between **two (2), three (3) or four (4)** typically, which allows the tool to start placing the datapoints indexed at 2, 3 or 4 into the test set. If the percentage of the test set (validation set) datapoints is fixed at 20%, the periodic calculation becomes 5 which allows the tool to include every 5<sup>th</sup> datapoint starting with index 2, 3 or 4 into the test set. By iterating through different splitting parameter values, it can be ensured that the test set is sampled differently each time, which can help assess the stability and generalization capability of your model. The approach of varying the splitting parameter (*sp*) to create different test sets is a robust method for evaluating machine learning models. It ensures comprehensive, unbiased, and representative testing, leading to more reliable and generalizable models.

#### *Random division*

Random division is another method to divide the dataset into training set and test set. This technique involves splitting the dataset into training set and test set by randomly placing datapoints into test set (validation set). For this technique to complete, the tool assigns indices to the datapoints, starting from 0 for better consistency and readability. Once the indices are assigned, the tool splits the data through into training and test sets from the whole dataset. This method is useful in creating test set and training set which has representation from the complete dataset. This offers a good mix of all the datapoints, leading to a balanced evaluation of the QSAR model. Finally, fixing the random state value for this technique ensures reproducibility of the training set and test set which is often required for generating stochastic model generation technique.

#### *KMCA (k-means cluster analysis)*

The final dataset division technique offered by SFS-QSAR is KMCA (*k-means cluster analysis*). *k-means Cluster Analysis* (KMCA) is a popular unsupervised machine learning algorithm used for partitioning a dataset into K clusters, where each data point belongs to the

cluster with the nearest mean. This technique is useful for discovering the underlying structure of the data and grouping similar data points together. In the first step, the number of clusters is fixed to identify the number of clusters required to group the complete data. In the next step, the K centroids (means of the clusters) are calculated and finally, the datapoints are placed into cluster which have the closest centroid value. The major disadvantage of this technique is that *k-means* assumes that clusters are spherical and equally sized, which might not always be the case.

The screenshot shows the SFS-QSAR application window. The 'Data preparation' tab is active. The 'Select Data' field contains the path 'oumya/Documents/SFS-QSAR-tool\_v2-main/dataset.csv'. The 'Variance cutoff' is set to 0.001. Under 'Dataset division techniques', the '%Data-points(validation set)' is 20. The 'Activity sorting' radio button is selected. The 'Start point' is 2. The 'Seed value' and 'Number of clusters' fields are empty. A 'Generate train-test sets' button is visible at the bottom.

For example, if a value 5 is set in this option, the data will be divided into 5 clusters on the basis of response variable and starting descriptors. The user needs to mention 'Seed value' (similar to random division) since from each cluster the test set compounds will be collected randomly.

For example, a dataset by the name of *dataset.csv* has been provided in the **github** ([https://github.com/ncordeirfcup/FXR\\_antagonists/tree/main/2DQSAR](https://github.com/ncordeirfcup/FXR_antagonists/tree/main/2DQSAR)) repository. The dataset has been placed in the main folder directory (please refer to the figure above to understand the location of the dataset in the system). The variance cut-off has been fixed at 0.001 and %Data-points (validation set) has been chosen as 20. Afterwards, the dataset division can be carried out by multiple options. Here, for easy understanding, the method chosen is Activity sorting with start point 2.

4. The next step involves generating training set and test set by clicking on the tab.

#### **b. Regression model development**

Once the dataset has been divided into training set and test set, now the user should go to the next tab.

The graphical interface for this tab looks like:

SFS-QSAR

Data preparation Regression model development Classification model development

Select training set  Browse

Select test/screening set  Browse

Do you have pretreated file: ☐ Yes ☒ No

Type output folder name

**Stepwise multiple linear regression**

Correlation cutoff  ☐ Y-randomization

Variance cutoff  Number of runs

Maximum steps  % of CV increment

Cross\_validation

Floating: ☐ True ☒ False

Scoring: ☒ R2 ☐ NMAE ☐ NMPD ☐ NMGD

Generate regression model

1. In the first tab, the user should select the training set generated within the directory of the tool while the generated test should be chosen in the next tab.
2. If the user has existing pre-treated file, then click on Yes or the data pre-treatment is carried out by keeping the existing setting.
3. The user should choose a suitable output folder name which is easier to locate.
4. For generating stepwise MLR (multiple linear regression) model, the user should keep **Correlation cutoff** as required as it refers to a threshold used to determine whether independent variables (features) should be included in the regression model based on their correlation with the dependent variable (target) or with each other. In MLR, managing multicollinearity through a correlation cutoff is crucial for improving model stability, interpretability, and predictive performance. By setting an appropriate threshold, the user can ensure that the regression model captures the essential relationships between variables while maintaining statistical rigor and computational efficiency. **For example**, the correlation cutoff is set at 0.99.
5. A **variance cutoff** refers to a threshold used to filter out features (independent variables) based on their variance or standard deviation. This approach is typically used to identify and remove features that have very low variance across the dataset. By setting a variance cutoff, the user can systematically identify and exclude features that do not significantly contribute to the predictive power of the model. Features with low variance may capture noise or random fluctuations rather

than meaningful patterns. Including such features can lead to overfitting, where the model learns from noise rather than signal. **For example**, the variance cutoff is fixed at 0.001.

6. In the next step, Maximum steps should be fixed by the user. Generally, in MLR, "maximum steps" refers to the maximum number of iterations or steps allowed in the stepwise regression process. It helps control the complexity of the model selection process, improves computational efficiency, and prevents overfitting by limiting the number of variables included in the final model. Adjusting the maximum steps parameter requires careful consideration of computational resources, statistical criteria, and domain knowledge to achieve optimal model performance. **For example**, the maximum steps kept here is 5.
7. The term "% CV increment" typically refers to the increase or decrease percentage in cross-validation (CV) metrics when evaluating different models or parameters. Cross-validation is a technique used to assess how well a model generalizes to an independent dataset by partitioning the data into subsets, training the model on some subsets, and evaluating it on the remaining ones. % CV increment measures the change in a cross-validation metric when a model or parameter changes. It helps in comparing different models or tuning parameters to determine which configuration performs better.
8. **Cross-validation (CV)** in Multiple Linear Regression (MLR) is a technique used to assess the performance and generalization ability of a regression model. It is particularly useful in MLR to evaluate how well the model predicts the outcome variable when applied to new data. The primary goal of cross-validation is to estimate how well the MLR model will generalize to an independent dataset that was not used during model training. The tool uses leave-one-out cross validation where each data point is used as a validation set in turn, with the rest of the datapoints in the dataset as training set. This is also known as K-fold cross validation where K is the number of datapoints in the dataset. By systematically validating the model across multiple subsets of the data, cross-validation helps in improving the reliability and effectiveness of MLR models. (**For example**, for 5-fold cross validation put 5)
9. The **Y randomization** test, also known as Y scrambling or response permutation testing, is a technique used to validate the robustness and significance of a predictive model. It is particularly useful for ensuring that a model is not simply overfitting the data but genuinely capturing the underlying relationship between the predictors (X) and the response variable (Y). The main goal of the Y randomization test is to check whether the model's performance is better than what would be expected by chance. This is done by comparing the model's performance on the original data to its performance on randomly shuffled data. (**For example**, put 1000 in the box to make 1000 iterations of the model)
10. "Floating Search Method" is used for selecting an optimal subset of features. This method aims



to balance between forward and backward selection steps to improve the feature selection process's efficiency and effectiveness. By keeping the Floating option **True**, the SFS feature selection techniques is advanced into SFFS (Sequential Floating Forward Selection) feature selection which allows dynamic addition of features, resulting in better performing feature subsets.

## 11. Scoring functions

- i.  $R^2$  (Determination coefficient)
- ii. NMAE (Negative mean Absolute error)
- iii. NMPD (Negative mean Poisson deviance)
- iv. NMGD (Negative mean gamma deviance)

The tool offers four distinct scoring functions to evaluate the statistical parameters of the regression model, the details of which is stored as **Results.txt** file in the output folder.

The output folder contains various statistical parameters in the **Results.txt** file, while correlation matrix, training set prediction, test set prediction, plot between observed value and predicted value can be found as separate files.

The screenshot displays the SFS-QSAR application window with the 'Regression model development' tab selected. The interface includes the following elements:

- Navigation Tabs:** Data preparation, Regression model development (active), Classification model development.
- Select training set:** A text field containing 'ents/SFS-QSAR-tool\_v2-main/dataset\_tr.csv' and a 'Browse' button.
- Select test/screening set:** A text field containing 'ents/SFS-QSAR-tool\_v2-main/dataset\_ts.csv' and a 'Browse' button.
- Pretreated file:** A radio button group with 'Yes' and 'No' (selected) options.
- Output folder:** A text field with 'output' entered.
- Stepwise multiple linear regression section:**
  - Correlation cutoff:** Text field with '0.99'.
  - Variance cutoff:** Text field with '0.001'.
  - Maximum steps:** A spinner box set to '5'.
  - % of CV increment:** Text field with '0'.
  - Y-randomization:** A checked checkbox.
  - Number of runs:** Text field with '1000'.
  - Cross\_validation:** A spinner box set to '0'.
  - Floating:** Radio buttons for 'True' (selected) and 'False'.
  - Scoring:** Radio buttons for 'R2' (selected), 'NMAE', 'NMPD', and 'NMGD'.
- Generate regression model:** A prominent orange button at the bottom.

The training set (**dataset\_tr.csv**) and test set (**dataset\_ts.csv**) are deposited in the main directory. Once the training and test sets are generated, they are selected by Browse tab in the interface. Due to unavailability of the pretreated file, it has been kept No. The **Correlation cutoff** has been kept at **0.99**

while *Variance cutoff* is kept at *0.001*. *Maximum steps* have been kept at *5* while *%CV increment* and *Cross\_validation* have been fixed at *0*. The number of iterations for *Y-randomization Number of runs* have been fixed as *1000*. The floating option should be kept as True while for the ease of understanding the *Scoring* is chosen as *R<sup>2</sup>*. Once the selections are completed, the Generate regression model tab has been applied which produces the *output* folder containing *Results.txt* file. The statistical parameters can be found in the file along with the other results from the regression model.

### c. Classification model development

The dataset preparation for classification-based model development is somewhat different than preparing dataset for regression model. In this particular model development method, the independent variable is classified into binary (i.e. a threshold value for the independent variable is fixed) e.g. 1 and 0.

Once the classification model development tab is clicked upon, the following graphical interface will open:

The screenshot shows the 'SFS-QSAR' application window with the 'Classification model development' tab selected. The interface includes the following elements:

- Navigation Tabs:** 'Data preparation', 'Regression model development', and 'Classification model development'.
- File Selection:** 'Select training set' and 'Select test set' text boxes, each with a 'Browse' button.
- Pretreated File:** 'Do you have pretreated file:' with radio buttons for 'Yes' and 'No' (selected).
- Output Folder:** 'Type output folder name' text box.
- Selection Method:** 'Increment based selection:' with radio buttons for 'True' and 'False' (selected), and a '% lambda decrease' text box.
- Parameters:**
  - 'Correlation cutoff' text box.
  - 'Variance cutoff' text box.
  - 'Maximum steps' spinner box set to 0.
  - 'FS-LDA' checkbox (unchecked).
  - 'SFS-LDA' checkbox (unchecked).
  - 'Y-randomization' checkbox (unchecked).
  - 'p-value to enter' text box.
  - 'Cross\_validation' spinner box set to 0.
  - 'p-value to remove' text box.
  - 'Floating:' radio buttons for 'True' and 'False' (selected).
  - 'Number of runs' text box.
  - 'Scoring:' radio buttons for 'Accuracy' and 'ROC\_AUC'.
- Action Buttons:** Two 'Generate model' buttons at the bottom.

As the dataset division is already executed in the first tab, here the user needs to put the training set and test set in the tabs respectively by browsing them in the main directory. Similar to regression model development, if the user has pretreated file, the correct option should be selected. A unique output folder name should be given by the user to identify the output file which is created in the directory.

1. The next step requires the user to decide whether increment-based selection should be used

during classification model development. Increment based selection can be a powerful tool in building efficient and effective classification models by focusing on the most relevant features. Increment based selection in classification models generally refers to a method where features are added incrementally to improve the model's performance. This approach can help in selecting the most relevant features while reducing the dimensionality of the dataset, thus potentially improving the model's predictive quality.

2. If the user chooses, Increment based selection as True, the “**% lambda decrease**” tab is activated which requires input in the form of numerical values. The “**% lambda decrease**” typically refers to a technique used in regularization methods in machine learning and statistical models. A “**% lambda decrease**” might refer to the process of gradually reducing the regularization strength during model training to allow for a more flexible model while still preventing overfitting. This technique can be part of a more comprehensive hyperparameter tuning strategy to find the optimal  $\lambda$  that balances bias and variance. For this particular dataset, **% lambda decrease** value has been fixed to **10**.
3. In the next step, the **Correlation cutoff** is fixed at **0.99**, **variance cutoff** is taken as **0.001** and **Maximum steps** have been fixed at **10**.
4. In the following step, the **FS-LDA** tab is selected and both **p-value to enter** and **p-value to remove** is fixed at **0.5**.
5. Following this, the user requires mentioning the number of iterations for **Y-randomization test**. In case of this example, the value has been fixed at **1000**.
6. Finally, the user needs to click ‘**Generate model**’ tab to get the linear classification model.
7. Once the model is generated, the outcomes can be found in the output folder of the directory.
8. The output folder contains four different contents, namely **1. correlation.csv**, **2. Prediction.csv**, **3. Results.txt** and **4. ROCplot.png**.

SFS-QSAR

Data preparation | Regression model development | **Classification model development**

Select training set: C:/Users/Soumya/Documents/SFS-QSAR-tool [Browse]

Select test set: C:/Users/Soumya/Documents/SFS-QSAR-tool [Browse]

Do you have pretreated file: ☐ Yes ☒ No

Type output folder name: output\_cm\_fslda

Increment based selection: ☒ True ☐ False % lambda decrease: 10

Correlation cutoff: 0.99 Variance cutoff: 0.001 Maximum steps: 10

☒ FS-LDA ☐ SFS-LDA ☒ Y-randomization

p-value to enter: 0.5 Cross\_validation: 0

p-value to remove: 0.5 Floating: ☐ True ☒ False Number of runs: 1000

Scoring: ☐ Accuracy ☐ ROC\_AUC

[Generate model] [Generate model]

For users who want to use the SFS-LDA algorithm for model development, may first select the tab by clicking on it. The previous fields may be populated as per the requirement of the user. For this particular study, the same requirements have been met as those for FS-LDA.

1. Once the SFS-LDA tab is clicked, the user needs to fix the Cross-validation value for the model. For this study, the cross-validation value is fixed at 5.
2. The floating function may be kept True which works similarly as mentioned previously during regression model generation.
3. This tool offers two different scoring features **1. Accuracy** and **2. ROC\_AUC**

❖ Accuracy is a common metric used to evaluate the performance of classification models. It is defined as the ratio of the number of correct predictions to the total number of predictions. Mathematically, accuracy can be expressed with the following equation:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

TP (True Positives): The number of correctly predicted positive samples.

TN (True Negatives): The number of correctly predicted negative samples.

FP (False Positives): The number of incorrectly predicted positive samples.

FN (False Negatives): The number of incorrectly predicted negative samples.

- ❖ The ROC curve is a graphical representation of a classifier's performance. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold levels. The TPR and FPR are defined as:

➤  $TPR = TP/(TP+FN)$

➤  $FPR = FP/(FP+TN)$

The AUC is the area under the ROC curve. It provides a single scalar value to summarize the model performance. The AUC value ranges from 0 to 1, where:

*AUC = 1 represents a perfect classifier.*

*AUC = 0.5 represents a classifier with no discriminative power (random guessing).*

*AUC < 0.5 suggests that the classifier is performing worse than random guessing.*

4. Finally, the user should click on '**Generate model**' tab to get the linear classification model and select the scoring feature accordingly.
5. Once the model is generated, the outcomes can be found in the output folder of the directory.
6. The output folder contains four different contents, namely **1. correlation.csv**, **2. Prediction.csv**, **3. Results.txt** and **4. ROCplot.png**.

The screenshot displays the SFS-QSAR software interface, specifically the 'Classification model development' tab. The window title is 'SFS-QSAR'. The interface includes several input fields and checkboxes for configuring the model development process. The 'Select training set' and 'Select test set' fields both point to 'C:/Users/Soumya/Documents/SFS-QSAR-tool', each with a 'Browse' button. The 'Do you have pretreated file:' section has radio buttons for 'Yes' and 'No', with 'No' selected. The 'Type output folder name' field contains 'output\_cm\_sfslda'. The 'Increment based selection' section has radio buttons for 'True' (selected) and 'False', and a '% lambda decrease' field set to '10'. The 'Correlation cutoff' field is '0.99', 'Variance cutoff' is '0.001', and 'Maximum steps' is '10'. The 'FS-LDA' checkbox is unchecked, while 'SFS-LDA' and 'Y-randomization' are checked. The 'p-value to enter' and 'p-value to remove' fields are empty. The 'Cross\_validation' field is '5', and the 'Number of runs' field is '1000'. The 'Floating' section has radio buttons for 'True' (selected) and 'False'. The 'Scoring' section has radio buttons for 'Accuracy' (selected) and 'ROC\_AUC'. At the bottom, there are two orange 'Generate model' buttons.

SFS-QSAR

Data preparation Regression model development Classification model development

Select training set C:/Users/Soumya/Documents/SFS-QSAR-tool Browse

Select test set C:/Users/Soumya/Documents/SFS-QSAR-tool Browse

Do you have pretreated file: ☐ Yes ☒ No

Type output folder name output\_cm\_sfslda

Increment based selection: ☒ True ☐ False % lambda decrease 10

Correlation cutoff 0.99 Variance cutoff 0.001 Maximum steps 10

☐ FS-LDA ☒ SFS-LDA ☒ Y-randomization

p-value to enter  Cross\_validation 5

p-value to remove  Floating: ☒ True ☐ False Number of runs 1000

Scoring: ☒ Accuracy ☐ ROC\_AUC

Generate model Generate model