Co-funded by the Horizon 2020 Framework Programme of the European Union

Grant agreement No. 671555

ExCAPE
Exascale Compound Activity Prediction Engines

Future and Emerging Technologies (FET)

Call: H2020-FETHPC-2014
Topic: FETHPC-1-2014
Type of action: RIA

Deliverable D3.3
Report: Criteria Report
ExCAPE benchmarking criteria report

Due date of deliverable: March. 01. 2016
Actual submission date: May. 05. 2016

Start date of Project: 1.9.2015 Duration: 36 months

Responsible Consortium Partner: JP
Contributing Consortium Partners: AZ, IDEA
Name of author(s) and contributor(s): Hongming Chen, Lars Carlsson, Ola Engkvist (AZ), Vladimir Chupakin (JP), Nina Jeliazkova (IDEA)
Revision: V8.0
Internal reviewer(s): Tom Vander Aa

Project co-funded by the European Union within the Horizon 2020 Framework Programme (2014-2020)		
Dissemination Level		
PU	Public	**X**
PP	Restricted to other programme participants (including the Commission Services	
RE	Restricted to a group specified by the consortium (including the Commission Services)	
CO	Confidential, only for members of the consortium (including the Commission Services)	

Document revision tracking

This page is used to follow the deliverable production from its first version until it has been reviewed by the assessment team. Please give details in the table below about successive releases.

Release number	Date	Reason of this release and/or validation	Dissemination
V1.0	March, 15th, 2016	The deliverable is related to Task 3.4 of WP3. It was discussed during the Ostrava meeting in March.	All project partners
V2.0	April, 24th, 2016	Modified based on the feedbacks obtained during Ostrava meeting.	All project partners
V3.0	May, 2nd, 2016	Modified based on internal discussion	All project partners
V4.0	May, 5th, 2016	Reformatting the document	All project partners
V5.0	July, 5th, 2016	Modified based on EU reviewers' comments	Public
V8.0	April 11, 2017	Internal reviewing	Public

Glossary

TN	true negative
TP	true positive
FN	false negative
FP	false positive
MCC	Matthews correlation coefficient
Log-loss	logarithmic loss
MSE	mean squared error
ROC curve	receiver operating characteristic curve
EC	expected cost
TPR	True Positive Rate
FPR	False Positive Rate
AUC	area under curve
ROI	return of investment
TEP	total expected profit
TEC	total expected cost
ROCCH	ROC convex hull

Link to Tasks

Task number	Work from task carried out	Deviations from task technical content
Task3.4	Development of a benchmarking methodology for evaluating the different algorithms	Task was carried out as expected

Table of contents

1 Executive summary

The main objective for this report is to propose a metric relevant to the industry partners of ExCAPE when evaluating the performance of different probabilistic prediction algorithms based on binary classification methods. The general situation is that after obtaining predictions for a set of compounds an arbitrarily sized subset of compounds will be further investigated. Thus, a suitable metric is log-loss based on the size of the subset and a given class label. Kappa value, as a robust classification performance measurement by taking into account the agreement occurring by chance, will be used as another criteria for ExCAPE. Furthermore, it is also of importance to understand what the computational complexity of any given algorithm is, both in terms of memory requirements and floating point operations as a function of training and prediction set size, respectively.

The reason for late submission was due to consultation among WP3 partners for modification.

2 Introduction- Aim

The choice of a proper performance metric for evaluating machine learning models is an old but still evolving area which has incorporated many different performance measurement methods along the way[1]. There are different metrics for the tasks of classification, regression, ranking, clustering, topic modelling, etc. Considering the heterogeneous nature of the big data in life science, the ExCAPE project mainly focuses on building classification models. This report tries to summarize some of the most common performance criteria used in literatures for benchmarking and comparing performance between different machine learning models.

Classification is about predicting class labels given input data. In binary classification, there are two possible output classes. In multi-class classification, there are more than two possible classes. We will mainly discuss about binary classification metrics, as they can be easily extended to multiple classes, and also recommend the criteria which are suitable for using in ExCAPE benchmarking study.

3 Predictive Performance Evaluation

3.1 Prediction Accuracy

Prediction accuracy is the most straight forward measurement for classification performance, which is simply the ratio between the number of correct predictions and the total number of predictions (the number of test data points).

$$Accuracy = \frac{\#\ of\ currect\ predictions}{\#\ of\ data\ points} \qquad (1)$$

3.2 Confusion Matrix

Prediction accuracy in equation 1 looks easy enough. However, it makes no distinction between classes; correct answers for class 0 and class 1 are treated equally. Sometimes this is not enough. For example you might want to look at how many examples failed for class 0 vs. class 1. Various additional classification metrics based on confusion matrix has been developed and is remain the most common way of measuring performance. In the field of machine learning, a confusion matrix[2], also known as a contingency table or an error matrix, is a specific table with two rows and two columns that reports the number of *false positives (FP)*, *false negatives (FN)*, *true positives (TP)*, and *true negatives (TN)*. This allows more detailed analysis than mere proportion of correct guesses (accuracy). In this table, x and $\overline{x}$ represent the events X= positive and X= negative and y and $\overline{y}$ represent the events Y = positive and Y = negative, respectively, meaning the actual/predicted

X \ Y	y	$\overline{y}$	
x	$TPos$	$FNeg$	Pos
$\overline{x}$	$FPos$	$TNeg$	Neg
	$PPos$	$PNeg$	N

The two inner rows correspond to actual classes, while the two inner columns correspond to predicted classes.

Figure 1. Illustration of confusion matrix

Many various measurements[3] can be derived from confusion matrix. Some very common ones have been listed in the Table 2. The sensitivity and specificity can be regarded as the per-class accuracy, while the accuracy in eq.1 reflects the percentage of correct prediction for all classes. Precision of positive and negative represent the probability of correct classification from prediction point of view. Fscore is a harmonic average for positive precision and sensitivity. The Kappa value is a statistic result which measures inter-rater agreement for class items. It is generally thought to be a more robust measure than simple percent agreement calculation, since the agreement occurring by chance is taken into account. The Matthews correlation coefficient (MCC) takes into account true and false positives and negatives and is

generally regarded as a balanced measure which can be used even if the classes are of very different sizes. Generally MCC and Kappa parameters are regarded as more balanced measurement than others and were widely used for classification measurement.

Table 2 Some common classification matrices based on confusion matrix

Measure	Formulae[a]	Description
Accuracy	$\frac{TP+TN}{TP+FP+FN+TN}$	probability to correctly classify compounds
Sensitivity (Recall, True Positive Rate)	$\frac{TP}{TP+FN}$	probability to predict positive when true class is positive
Specificity (True Negative Rate)	$\frac{TN}{FP+TN}$	probability to predict negative when true class is negative
Positive precision	$\frac{TP}{TP+FP}$	probability to correctly classify compounds predicted to be positive
Negative precision	$\frac{TN}{FN+TN}$	probability to correctly classify compounds predicted to be negative
Fall-out (False Positive Rate)	$\frac{FP}{FP+TN}$	Probability of negatives being predicted as positive
Fscore	$\frac{2*Sensitivity*Pos.precision}{Sensitivity+Pos.precision}$	harmonic average of positive precision and sensitivity.
Kappa	$\frac{Accuracy-Cp}{1-Cp}, where\ Cp = \frac{(TP+FP)*(TP+FN)+(TN+FN)*(TN+FP)}{(TP+FP+FN+TN)^2}$	true accuracy which is corrected by probability to obtain agreement by chance
Matthews Correlation Coefficient (MCC)	$\frac{(\mathrm{TP}*\mathrm{TN}-\mathrm{FN}*\mathrm{FP})}{\sqrt{(\mathrm{TP}+\mathrm{FN})(TP+FP)(TN+FN)(TN+FP)}}$	Correlation coefficient between the observed and predicted binary classifications

Note: a) TP: true positive, FP: false positive, TN: true negative, FN: false negative

3.3 Probabilistic predicting and scoring rules

Sometimes model prediction value is not a binary value such as “0” or “1”, but a probabilistic prediction instead. For example, a weather forecasting on raining is usually a probability value not the binary value “0” or “1”. In this case, scoring rules or scoring function is introduced to measure the accuracy of the probabilistic predictions. A score can be thought of as either a measure of the "calibration" of a set of probabilistic predictions, or as a "cost function" or “loss function”.

3.3.1 Log-loss

Log-loss, or logarithmic loss, gets into the finer details of a classifier. In particular, if the raw output of the classifier is a numeric probability instead of a class label of 0 or 1, then log-loss can be used. The probability essentially serves as a gauge of confidence. If the true label is 0 but the classifier thinks it belongs to class 1 with probability 0.51, then the classifier would be making a mistake. But it's a near miss because the probability is very close to the decision boundary of 0.5. Log-loss is a "soft" measurement of accuracy that incorporates this idea of probabilistic confidence.

Mathematically, log-loss for a binary classifier looks like this:

$$log - loss = -\frac{1}{N}\sum_{i=1}^{N}[y_i log p_i + (1 - y_i)\log(1 - p_i)] \qquad (2)$$

Here p_i is the probability that the *i*-th data point belongs to class 1, as judged by the classifier. y_i is the true label and is either 0 or 1. This definition is actually tied to the concept of cross entropy in information theory: The cross entropy between two probability distributions over the same underlying set of events measures the average number of bits needed to identify an event drawn from the set. It can be regarded as describing the similarity of two probability distributions over the same underlying set of data, i.e. the distribution of the true labels and the predictions. It is very closely related to what's known as the relative entropy, or Kullback-Leibler divergence. Entropy measures the unpredictability of something. Cross entropy incorporates the entropy of the true distribution, plus the extra unpredictability when one assumes a different distribution than the true distribution. So log-loss is an information-theoretic measure to gauge the "extra noise" that comes from using a predictor as opposed to the true labels. By minimizing the cross entropy, we maximize the accuracy of the classifier.

3.3.2 Brier Score

The Brier score[4] is a proper score function that measures the accuracy of probabilistic predictions. It is applicable to tasks in which predictions must assign probabilities to a set of mutually exclusive discrete outcomes. The set of possible outcomes can be either binary or categorical in nature, and the probabilities assigned to this set of outcomes must sum to one (where each individual probability is in the range of 0 to 1). The most common formulation of the Brier score is:

$$BS = \frac{1}{N}\sum_{t=1}^{N}(f_t - O_t)^2 \qquad (3)$$

In which f_t is the probability that was forecast, O_t the actual outcome of the event at instance t (0 if it does not happen and 1 if it does happen) and N is the number of predicting instances. In effect, it is the mean squared error (MSE) of the forecast. This formulation is mostly used for binary events (for example "rain" or "no rain"). A better model is obtained by minimizing

the Brier score. If a multi-category forecast is to be evaluated, then the original definition given by Glenn W. Brier below should be used.

$$BS = \frac{1}{N}\sum_{t=1}^{N}\sum_{i=1}^{R}(f_{ti} - O_{ti})^2 \qquad (4)$$

4 Graphical Representation of Predictive Performance

4.1 Cost sensitive learning

For binary classification, two types of errors may occur: false positives and false negatives. Most learning systems deal with these errors as equally costly and try to minimize the overall error rate. Furthermore, a cost-sensitive learning system can be used in applications where the misclassification costs are known. A misclassification cost is simply a value that is assigned as a penalty for making a mistake. In this case, misclassification costs can be used in substitution for the error rate (accuracy), and a cost-sensitive learning system attempts to reduce the cost of misclassified examples instead of classification errors. Usually, a cost matrix is used to define the costs associated to a domain. A cost matrix is similar to a contingency table. If the values on the main diagonal are represented with negative costs, then these values can be interpreted as gains or profits. Each entry of a cost matrix defines a constant cost/profit for each type of error/hit that can be made by a classifier. Given a contingency table and a cost matrix, the expected cost, EC, can be computed using:

$$EC = \sum_{X \in \{x, \overline{x}\}} \sum_{Y \in \{y, \overline{y}\}} p(X,Y)c(X,Y) \qquad (5)$$

where $p(X,Y)$is the corresponding cell in the contingency table divided by N and $c(X,Y)$ is the cost/profit for that type of classification type of classification.

4.2 ROC Curve

There are also many graphical representations and tools for model evaluation, the most common one is probably the receiver operating characteristic (ROC) curve.[5] ROC curve is created by plotting sensitivity (True Positive Rate, TPR) against the False Positive Rate (FPR). It is a graphic tool to illustrate the performance of binary classifier. An example of ROC curve is shown in Figure 1

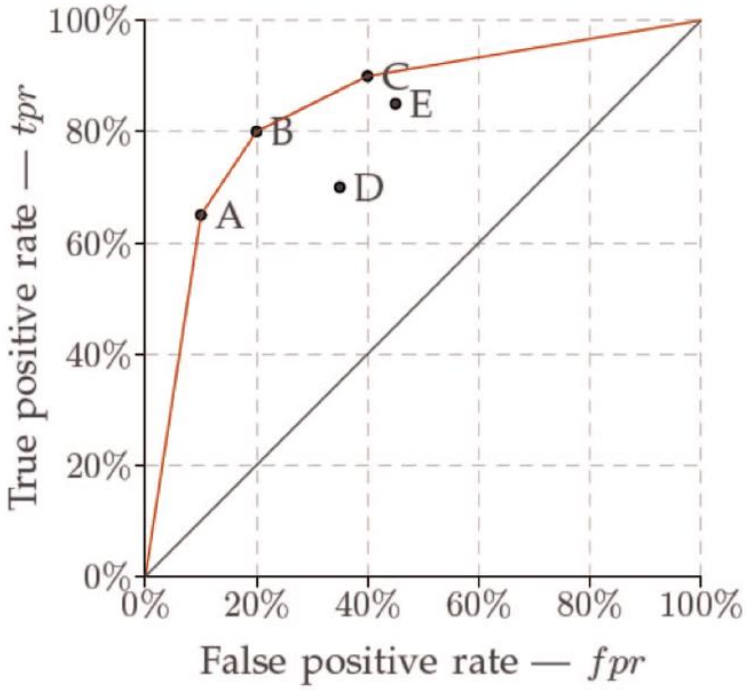


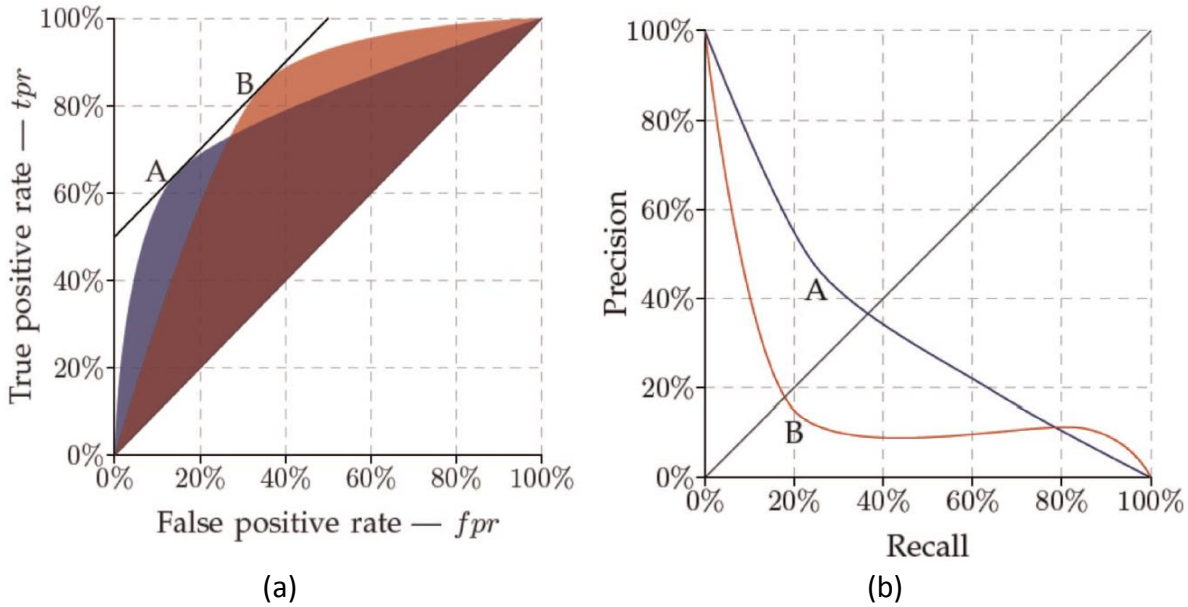
Figure 1. Illustration of ROC curve[6].

In ROC curve, the diagonal line represent the performance of random predicting model and curves sit above the diagonal line corresponding to models performing better than random (non-discrimination line) and vice versa. An example of ROC curve is shown in Figure 2, where A,B,C,D and E represents five different classifiers. The more the curve is close to the upper left corner, the better model performs and the more to the lower right corner, the worse the model performs. In binary system, the model prediction value is often a continuous score value S. With a given threshold T and certain decision rule, the model performance metrics TPR and FPR can be obtained. The ROC curve can be generated by varying the T value. ROC analysis provides tools to select possibly optimal models and to discard suboptimal ones independently from (and prior to specifying) the cost context or the class distribution. An example of ROC curve is shown in Figure 1, where A, B, C, D and E represents five different classifiers. The convex hull of the set of points in the ROC space is called the ROC convex hull (ROCCH) of the corresponding set of classifiers. A, B, C classifiers are on the convex hull, while D,E classifiers are not and therefore are suboptimal. However, the choice among A, B, and C depends on information regarding operational conditions.

Several statistical parameters derived from ROC curve can be used for measuring performance. For example: Youden's index[7] is defined as the intercept of the ROC curve with the line at 90 degrees to the no-discrimination line. Its value ranges from 0 to 1, and a zero value means a non-discriminative model and a value of 1 indicates that there are no false positives or false negatives, i.e. a perfect model. AUC (area under curve) is a very common parameter used in conjunction with ROC curve. Mathematically it is equal to the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one (assuming 'positive' ranks higher than 'negative').

4.3 Precision-Recall Curves

Precision-recall curves[8] are often used in information retrieval applications to evaluate ranked retrieval performance results. This is because information retrieval tasks are often characterized by a large skew in the class distribution, i.e., the number of negative cases heavily outnumbers the number of positive cases. There is similar scenario in high throughput screening (HTS) data in pharmaceutical industry and furthermore retrieving the active compounds (positive class) is of more interest than the inactive ones (negative class). This particular characteristic of information retrieval tasks makes the area of interest in a ROC graph compressed to a small corner in the lower left side of the ROC space.



(a) (b)

Figure 2. comparison of (a) ROC curve with (b) Precision-Recall curve[6]

In Precision-Recall curves, the y axis is the precision and the x axis is the recall. The objective of this axis setup is to make differences in the area of interest clearer than in the ROC space. Figure 2 shows the precision-recall corresponding to the ROC curves shown in Figure 1, for an arbitrarily chosen prevalence of positives of 5%. Analysing the curves, in spite of both models having the same AUC, we can see that model A is better at identifying positives than negatives (has higher precision) in almost all the x-axis. Furthermore, due to the low prevalence of positives, this difference is much more significant in the precision-recall space than in the ROC space. The higher the prevalence of the positives, the closer the curves in the precision-recall space are. This is because it is easy to obtain high precision in domains where the prevalence of positives is also high.

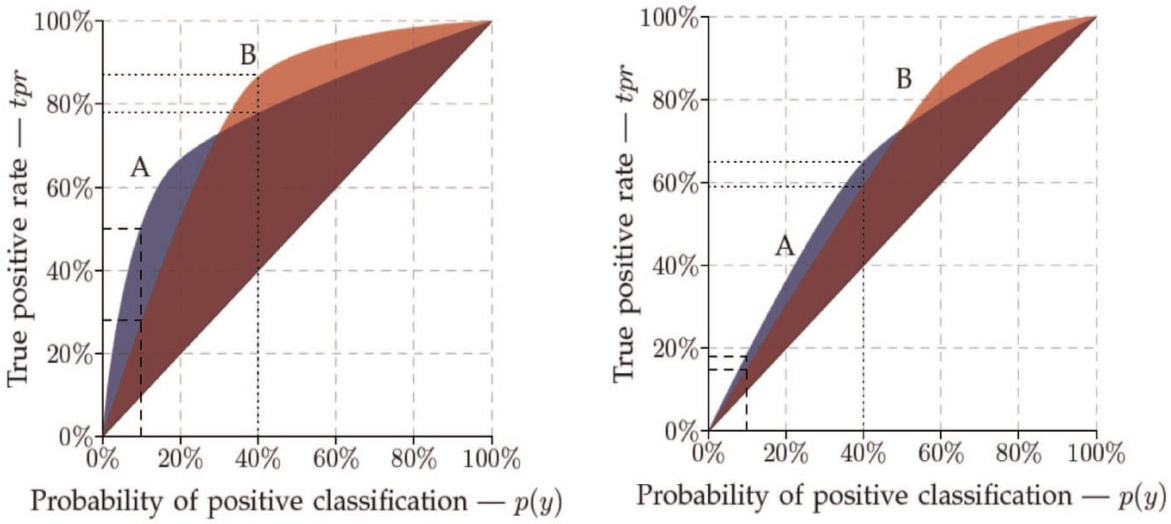


Figure 3. Lift curve at (a) 5% positive prevalence level and (b) 50% positive prevalence level[6]

4.4 Lift Graph

Similar to ROC curves, lift graphs[9] use the true positive rate (TPR) with the y-axis. However, the prevalence for positive class (proportion of positives in the total population) is associated to the x-axis instead of the false positive rate (FPR). This change makes lift graphs sensitive to operational conditions, i.e. the level of p(x). Since

$$p(y) = p(y|x)p(x) + p(y|\overline{x})p(\overline{x}) \qquad (6)$$

Therefore, $p(y)$ can be derived from $p(y|x)$ (TPR) and $p(y|\overline{x})$ (FPR) and the positive ($p(x)$) and negative ($p(\overline{x})$) class prevalence. Figure 3 shows lift curves at different positive prevalence ($p(x)$).

As with ROC graphs, a crisp classifier corresponds to a point in a lift graph. However, a set of points can be generated by varying the percentage of cases classified as positive. A lift curve is defined as the convex hull of all points generated. Lift graph can be useful to compare performance of models giving continuous score value. For example in chemo-informatics area, it was often used to compare accuracy of different molecular similarity indexes or docking scoring functions in retrieving active compounds from large compound database.[10]

4.5 ROI Graph

Return of investment[11] (ROI) graphs are similar to lift graphs. However, ROI graphs associate the total expected profit (TEP), given by (7), to the y-axis

$$TEP = N \sum_{X \in \{x, \overline{x}\}} \sum_{Y \in \{y, \overline{y}\}} p(X,Y)c(X,Y)p(X) \qquad (7)$$

where N is the sample size, $p(X, Y)$ is the corresponding cell in the contingency table divided by N, is $c(X, Y)$ is the cost/profit for that type of classification and $p(X)$ represents the class prevalence. In order to compute the TEP, $c(X, Y)$ should associate positive values with profits and negative values with costs. An association of negative values with profits and positive values with costs changes to calculate the total expected cost (TEC). ROI graphs are limited to domains in which costs and class prevalence are constant and can be estimated confidently in advance.

5 Predictive Performance Criteria of ExCAPE

The ExCAPE project aims to utilize Big chemogenomics data existed in both public domain and pharma industry to build models for compound activity prediction. The state of the art scalable algorithms and implementations thereof suitable for running on future Exascale machines will be developed and applied on real world dataset in pharma industry. For any machine learning algorithms, industry scale chemogenomics dataset represents a big challenge due to following reasons:

- **Large and unbalanced data**: Most of the Chemogenomics data in the public domain is in small datasets mainly composed of active compounds, while in industry datasets are big and highly imbalanced with a prevalence of inactive compounds.
- **Biological data is complex and noisy**: Every response of a living organism to its environment depends on many factors, thus all biological and pharmacological data is noisy with a significant amount of false positive/negative data.
- **Significantly large feature space**: Although there is no consensus on how compounds should be encoded for a perfect model, substructure based fingerprints are commonly used as input features and the number of input feature could be tens of thousands.

Due to these features of the chemogenomics dataset, building binary classification model, instead of continuous model, for each protein target is the main goal for ExCAPE project. There is no consensus in machine learning community on which performance criteria is the optimal measurement to use. Considering the highly unbalanced dataset that we are facing in ExCAPE project, among the confusion matrix based criteria, Kappa value is chosen as one of the criteria to use due to its capability of dealing the unbalanced data while also taking into account the agreement occurring by chance. It has been widely used in chemo-informatics area.[12,13] For models generating probabilistic estimation, Log-loss will be used as the model criteria evaluating prediction performance. As there will be a limitation on the number of compounds that will be further investigated, the interest would be to assess the top M probable compounds for a certain activity. Thus Log-loss as stated in Equation 2 will become

$$log - loss_M = -\frac{1}{M}\sum_{i=1}^{M}[y_i log p_i + (1 - y_i)\log(1 - p_i)] \qquad (8)$$

where the index is assumed to represent the sorted p_i based on the class of interest. The number of compounds of interest would be specific to a certain prediction task and would have to be defined for each data set, individually.

In an industry setting, it is also of great value to understand what resources are required to predict a certain property given a training set and a prediction set. It is therefore of great value to understand what computational resources are needed and what the expected calculation time would be given those resources as a function of training and prediction set sizes. A proposition is that any suggested ExCAPE algorithm should be analysed so that these properties of the algorithm are well understood.

6 Conclusion

Predictive performance evaluation is a fundamental issue in development classification models and comparison of various prediction models. In current report, several major classification performance metrics were described. This includes the simple and convenient confusion matrix based classification criteria and more illustrative graphical metrics. In the end, two criteria specific for the tasks in ExCAPE project were described.

7 References

[1] D. J. Hand, Construction and Assessment of Classification Rules, John Wiley & Sons Inc, 1997.

[2] S. V. Stehman, Selecting and interpreting measures of thematic classification accuracy,Remote Sensing of Environment, 62 (1), 1997, 77–89

[3] *E. P. Costa, A. C. Lorena, Carvalho, A. A. Freitas*, A review of performance evaluation measures for hierarchical classifiers, AAAI Workshop, Vancouver, 2007

[4] G. W. Brier *Verification of Forecasts Expressed in Terms of Probability,Monthly Weather Review, **78**,* 1950, 1–3

[5] D. M. W. Powers, *Evaluation: From Precision, Recall and F-Measure to ROC, Informedness, Markedness & Correlation,* Journal of Machine Learning Technologies*, **2** (1), 2011, 37–63*

[6] R. C. Prati, G. Batista, M. C. Monard, A Survey on Graphical Methods for Classification

Predictive Performance Evaluation, IEEE Transactions on Knowledge, 23(11), 2011, 1601-

1618.

[7] W. J. Youden, Index for rating diagnostic tests, Cancer, **3**, 1950, 32–35.

[8] K. H. Brodersen, C. S. Ong, K. E. Stephan, J. M. Buhmann, The binormal assumption on precision-recall curves, Proceedings of the 20th International Conference on Pattern Recognition, 2010, 4263-4266

[9] C.X. Ling, C. Li, Data Mining for Direct Marketing: Problems and Solutions, Proc. Fourth Int'l Conf. Knowledge Discovery and Data Mining, 1998, 73-79

[10] Z. Zhou, A. K. Felts, R. A. Friesner, R. M. Levy, Comparative Performance of Several Flexible Docking Programs and Scoring Functions: Enrichment Studies for a Diverse Set of Pharmaceutically Relevant Targets, J Chem Inf Model. 47(4), 2007, 1599–1608.

[11] FYI On ROI: A Guide To Calculating Return On Investment , http:// http://www.investopedia.com/articles/basics/10/guide-to-calculating-roi.asp, (accessed on July 5th, 2016)

[12] S. Varadharajan, S. Winiwarter, L. Carlsson, O. Engkvist, A. Anantha, T. Kogej, M. Fridén, J. Stålring, H. Chen, Exploring *In Silico* Prediction of the Unbound Brain-to-Plasma Drug Concentration Ratio: Model Validation, Renewal, and Interpretation, *J. Pharm. Sci.*, 2015, 104, 1197-1206.

[13] H. Chen, S. Winiwarter, M. Friden, M. Antonsson, O. Engkvist, *In silico* prediction of unbound brain-to-plasma concentration ratio using machine learning algorithms, *J. Mol. Graph. Model.*, 2011, 29, 985-995.