Mathematical Underpinnings of Machine Learning

Project Report - Topic A

Authors: Izabela Telejko, Grzegorz Zbrzeżny May 2024

1 Introduction

In our project we wanted to compare various approaches to feature selection task. We implemented the following methods using information theory measures:

- JMIM (Joint Mutual Information Maximisation) [2],
- IGFS (Interaction Gain Feature Selection) [3],
- CMIM (Conditional Mutual Information Maximization) [4].

Moreover we used two arbitrary methods for the comparison, namely:

- Feature Selection based on AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion),
- Sequential L1-based Feature Selection.

The entire codebase for this project is stored in a public repository[1].

2 Methods overview

2.1 Mutual Information based methods

In this subsection we present our MI-based feature selection methods more indepthly. Note that every method is an iterative procedure in which in every step we chose one, best feature. In the equations for each method we assign X_{method_name} as candidate feature for a given step. Apart from that we denote:

- F as full set of predictors,
- S as set of all predictors selected before current step,
- $I(X;Y)$ as mutual information between X and Y,
- $I(X;Y|Z)$ as mutual information between X and Y given Z,
- $I(X,Y;Z) = I(Y;Z) + I(X;Z|Y)$,
- $I(X;Y;Z) = I(X,Y;Z) - I(X;Z) - I(Y;Z)$.

In the following subsections we present equations defining how to choose best feature in a given step for each method.

[1] https://github.com/izabelatelejko/2024L-MUML-Project

2.1.1 Joint Mutual Information Maximisation

First step:

$$X_{JMIM} = \underset{X_i \in F}{argmax}\ I(X_i; Y)$$

Every next step:

$$X_{JMIM} = \underset{X_i \in F/S}{argmax}[\underset{X_s \in S}{min}\ I(X_i, X_s; Y)]$$

2.1.2 Interaction Gain Feature Selection

$$X_{IGFS} = \underset{X_i \in F/S}{argmax}[I(X_i; Y) + \frac{1}{|S|} \sum_{X_s \in S} I(X_i; X_s; Y)]$$

2.1.3 Conditional Mutual Information Maximization

$$X_{CMIM} = \underset{X_i \in F/S}{argmax}\{I(X_i; Y) - \underset{X_s \in S}{max}[I(X_i; X_s) - I(X_i; X_s|Y)]\}$$

2.2 Other methods

In addition to the three mutual information-based methods, we utilized two alternative approaches for comparison. The first approach was forward feature selection using the Bayesian Information Criterion (BIC). This method involves iteratively adding features until the addition of another feature no longer improves the BIC. The second approach employed forward feature selection with L1 regularization. Starting with no features, this method iteratively selects one feature that maximizes the cross-validated score when a Logistic Regression model with L1 penalty is trained on the current set of selected features plus the new feature. This process continues until adding a new feature does not improve the model score by more than 0.001.

3 Stopping rule

For the methods which do not include Mutual Information we used default stopping rules as mentioned in the previous section. Whereas for the MI-based feature selectors we defined the custom stopping rule. Namely, after each iteration we calculate a custom score based on estimation of the $I(Y, X_i; S)$ as proposed by Battiti [1], but mutual information was replaced with normalized mutual information. If the score is lower than empirically chosen threshold equal to 0.03 we stop the algorithm. The custom stopping score is defined as it follows:

$$score(X_i) = NMI(X_i; Y) - \frac{1}{|S|} \cdot \sum_{X_s \in S} NMI(X_i; X_s)$$

where:

- X_i – candidate feature in step i,
- $H(X)$ – entropy of the X variable,
- $NMI(X; Y) = \frac{I(X;Y)}{min[H(X),H(Y)]}$ – normalized mutual information.

4 Datasets

In this section, we provide brief descriptions of each dataset used in our experiments. Table 1 summarizes the key information about each dataset after preprocessing.

4.1 Artificial Data

The synthetic data X, Y are generated according to the following formulas:

$$
\begin{aligned}
X_1 &\sim \mathcal{N}(0,1) \quad (\text{size: } n \times n_{\text{rel}}) \\
X_2 &\sim \mathcal{N}(0,1) \quad (\text{size: } n \times n_{\text{irrel}}) \\
X_3 &= X_1 + \mathcal{N}(0,0.1) \quad (\text{size: } n \times n_{\text{rel}}) \\
X_4 &= \text{interactions}(X_1) \quad (\text{size: } n \times \frac{n_{\text{rel}} \times (n_{\text{rel}}-1)}{2}) \\
X &= [X_1, X_i], \quad i \in \{2,3,4\} \\
Y_{\text{num}} &= X_1 \times [\beta_1, \beta_2, \ldots, \beta_{n_{\text{rel}}}]^T + \beta_0 \\
Y &= \text{qcut}(Y_{\text{num}}, n_{\text{classes}}),
\end{aligned}
$$

where:

- n: Number of rows to generate.
- n_{rel}: Number of relevant features.
- n_{irrel}: Number of irrelevant features.
- $betas = [\beta_0, \beta_1, \ldots, \beta_{n_{\text{rel}}}]$: List of coefficients for calculating the target variable.
- n_{classes}: Number of classes in the target variable.

This way we generated 3 different datasets, where X_1 are relevant features (n_{rel} features) and X_i are irrelevant features (either features of the same distribution as X_1, X_1 features with some noise added, or second order interactions of X_1).

4.1.1 XOR dataset

We also aimed to identify a dataset where MI-based methods might not perform well. To this end, we generated three features from a Gaussian normal distribution and created a target variable based on the logical XOR of the last one feature and the XOR of the first two features. Additionally, we included ten irrelevant features, also generated from a Gaussian normal distribution.

4.2 Real-world Data

4.2.1 Divorce dataset[2]

This dataset containing 150 observations was created basing on a study focused on predicting divorce of Gottman couples therapy. It consists of 54 predictors (answers for 54 questions scaled from 0 to 4) and one binary target variable (married or divorced). For our experiment purposes we did not need to perform any preprocessing to make this dataset viable for model training.

4.2.2 AIDS classification dataset[3]

This dataset originally published in 1996 contains 22 healthcare statistics and binary information about patients whether they have been diagnosed with AIDS or not. It has 2139 observations included. In case of this dataset we also did not have to perform any preprocessing.

[2] https://www.kaggle.com/datasets/rabieelkharoua/split-or-stay-divorce-predictor-dataset
[3] https://www.kaggle.com/datasets/aadarshvelu/aids-virus-infection-prediction

4.2.3 LoL Diamond FF15 dataset[4]

This dataset contains 88 game metrics of the first 15 minutes of approximately 20,000 matches along with binary target variable indicating whether blue or red team won the whole match. For the preprocessing we removed column with match index, since it was unique for each row.

4.2.4 Cancer dataset[5]

This dataset contains the characteristics of patients diagnosed with cancer. It has 31 predictors (visual characteristics of the cancer) and binary target variable which indicates which cancer type given patient had. As for the preprocessing we removed patient ID and one column containing only NaN values. Moreover we decoded target variable from M and B values to 0 and 1.

4.3 Gait classification[6]

This data set was created by calculating the walking parameters of a total of 16 different volunteers. It contains 48 observations along with 321 features. In case of the preprocessing we removed one observations which contained NaN values.

Dataset	obs.	feat.	relevant feat.
aids	2139	22	-
cancer	569	310	-
divorce	170	54	-
gait	47	321	-
lol	21515	86	-
generated_0	1000	35	5
generated_1	1000	10	5
generated_2	1000	15	5
xor	1000	13	3

Table 1: Datasets basic information after preprocessing.

5 Results evaluation

To compare the feature selection methods, we first applied each of them to our test datasets. Subsequently, we trained three selected models on the feature sets identified by each method for each dataset. In Figure 1, we showed the number of features chosen for each dataset and method, and for the generated data, we also indicated how many features were truly relevant. Moreover in Table 2 we presented how many relevant and irrelevant features were chosen for synthetic data. Additionally, we presented accuracy scores for K-Nearest Neighbors (Figure 2), Random Forest Classifier (Figure 3), and Support Vector Classification (Figure 4) models, trained on features selected by each method for each dataset.

[4] https://www.kaggle.com/datasets/jakejoeanderson/league-of-legends-diamond-matches-ff15
[5] https://www.kaggle.com/datasets/erdemtaha/cancer-data
[6] https://archive.ics.uci.edu/dataset/604/gait+classification

Dataset	# Selected	BIC	AIC	CMIM	JMIM	IGFS	L1
generated_0	Rel. Feat.	5	5	5	5	5	5
	Irrel. Feat.	0	6	0	0	0	3
generated_1	Rel. Feat.	5	5	5	4	5	5
	Irrel. Feat.	2	2	0	0	0	0
generated_2	Rel. Feat.	5	5	5	5	5	5
	Irrel. Feat.	0	2	0	0	0	2
xor	Rel. Feat.	0	0	1	0	0	0
	Irrel. Feat.	1	3	0	1	1	1

Table 2: Number of selected relevant and irrelevant features for synthetic datasets.

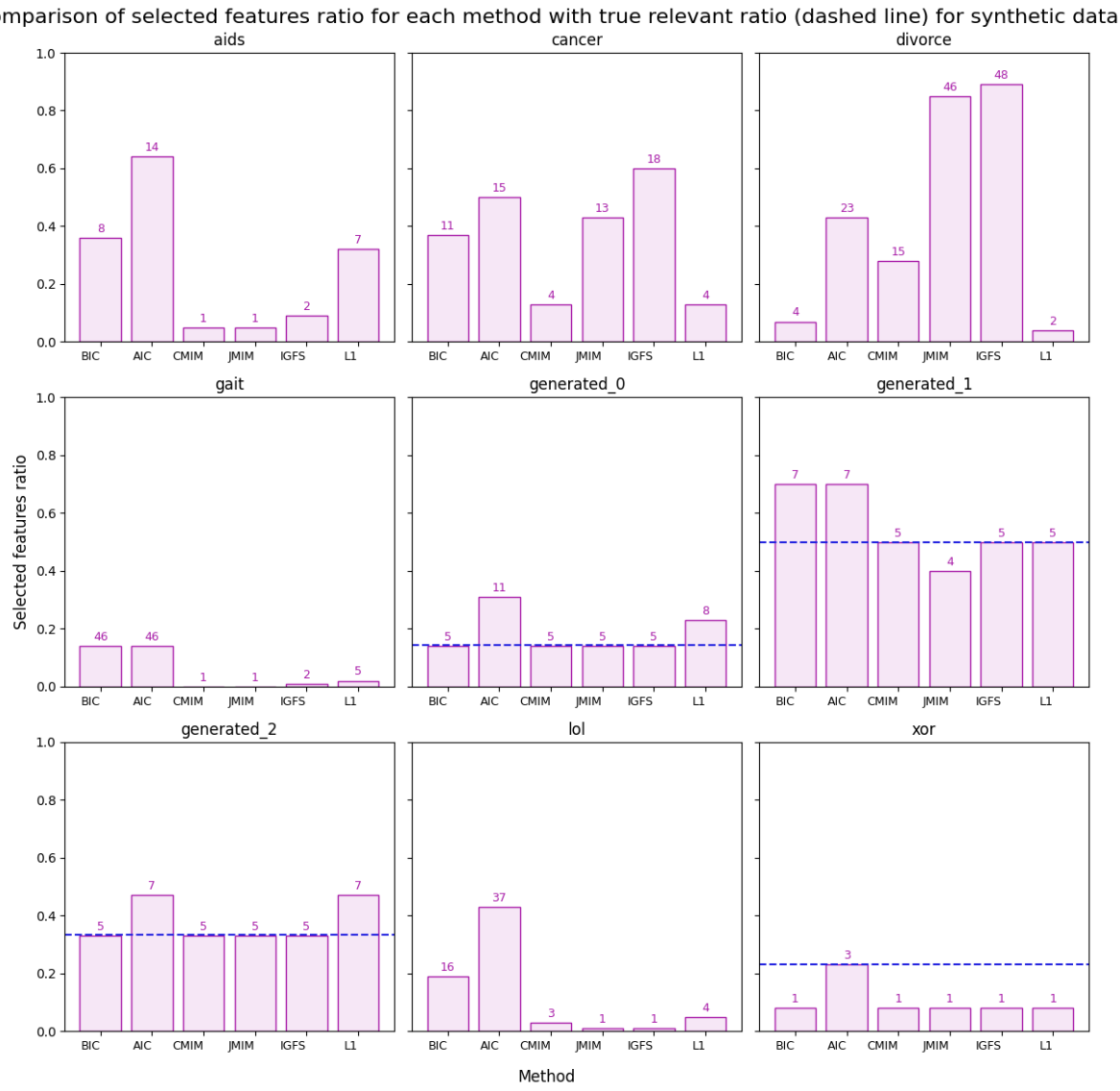

Figure 1: Number of selected features for each method and dataset.

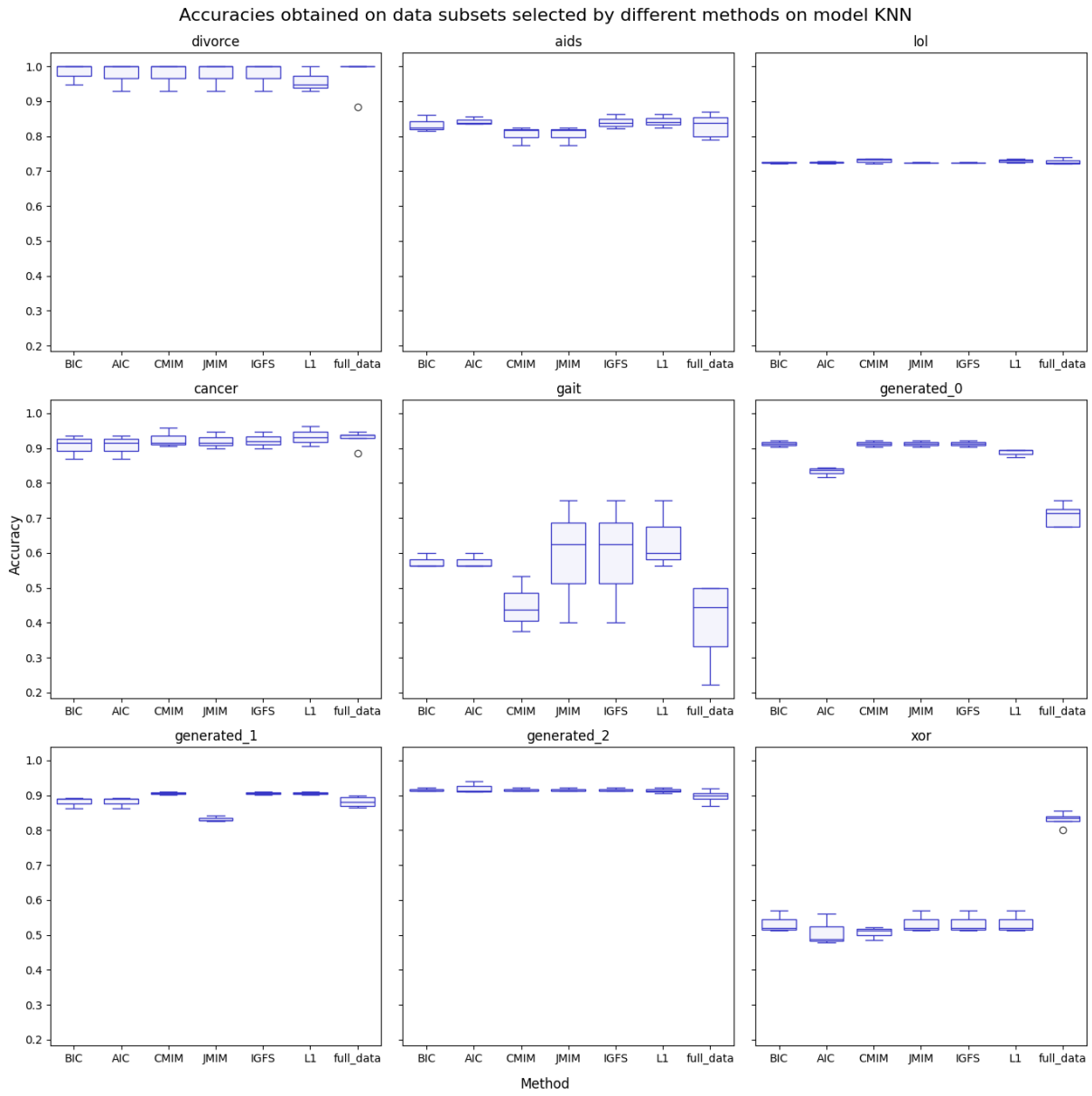

Figure 2: Accuracy scores for KNN models.

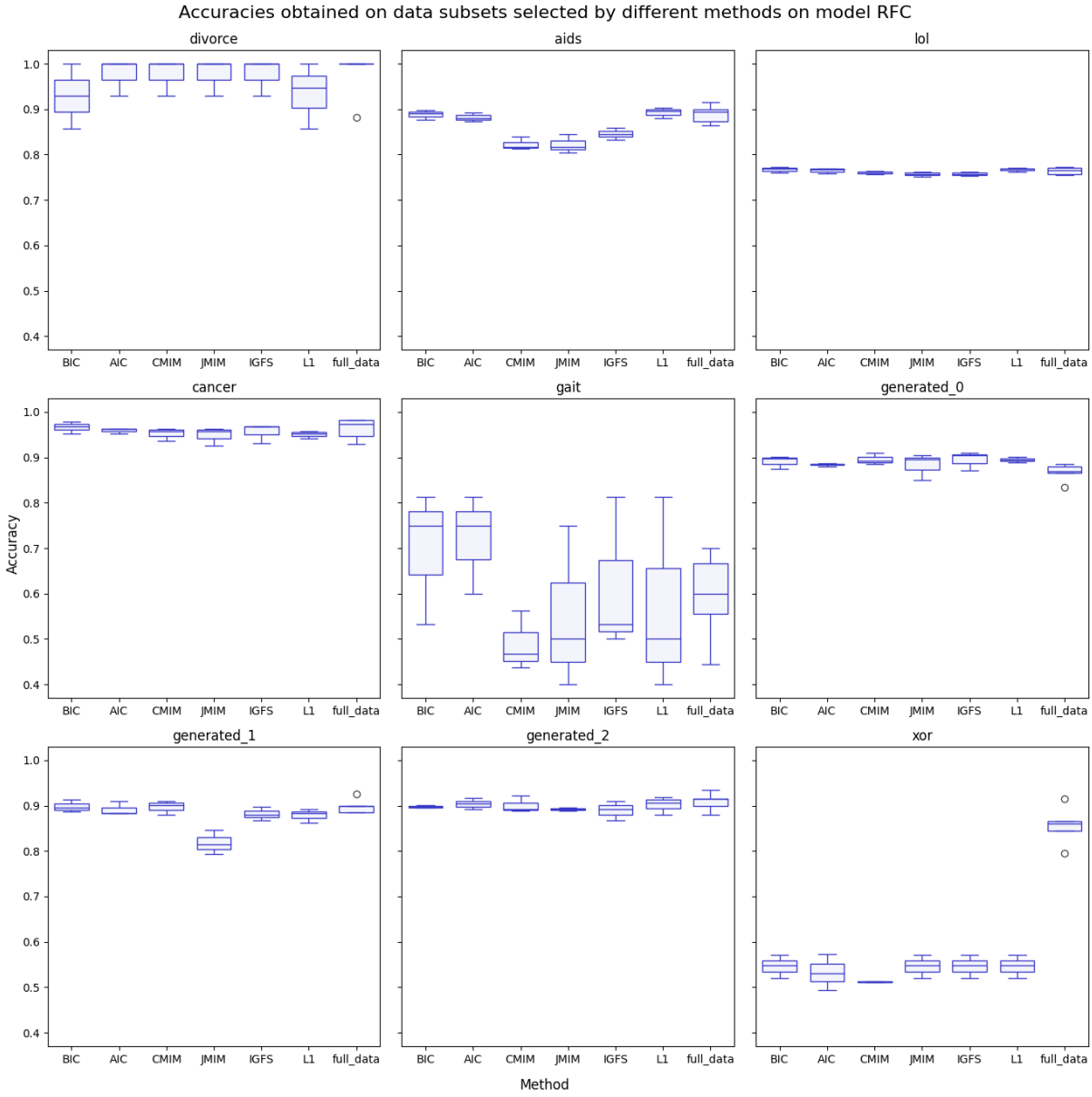

Figure 3: Accuracy scores for RFC models.

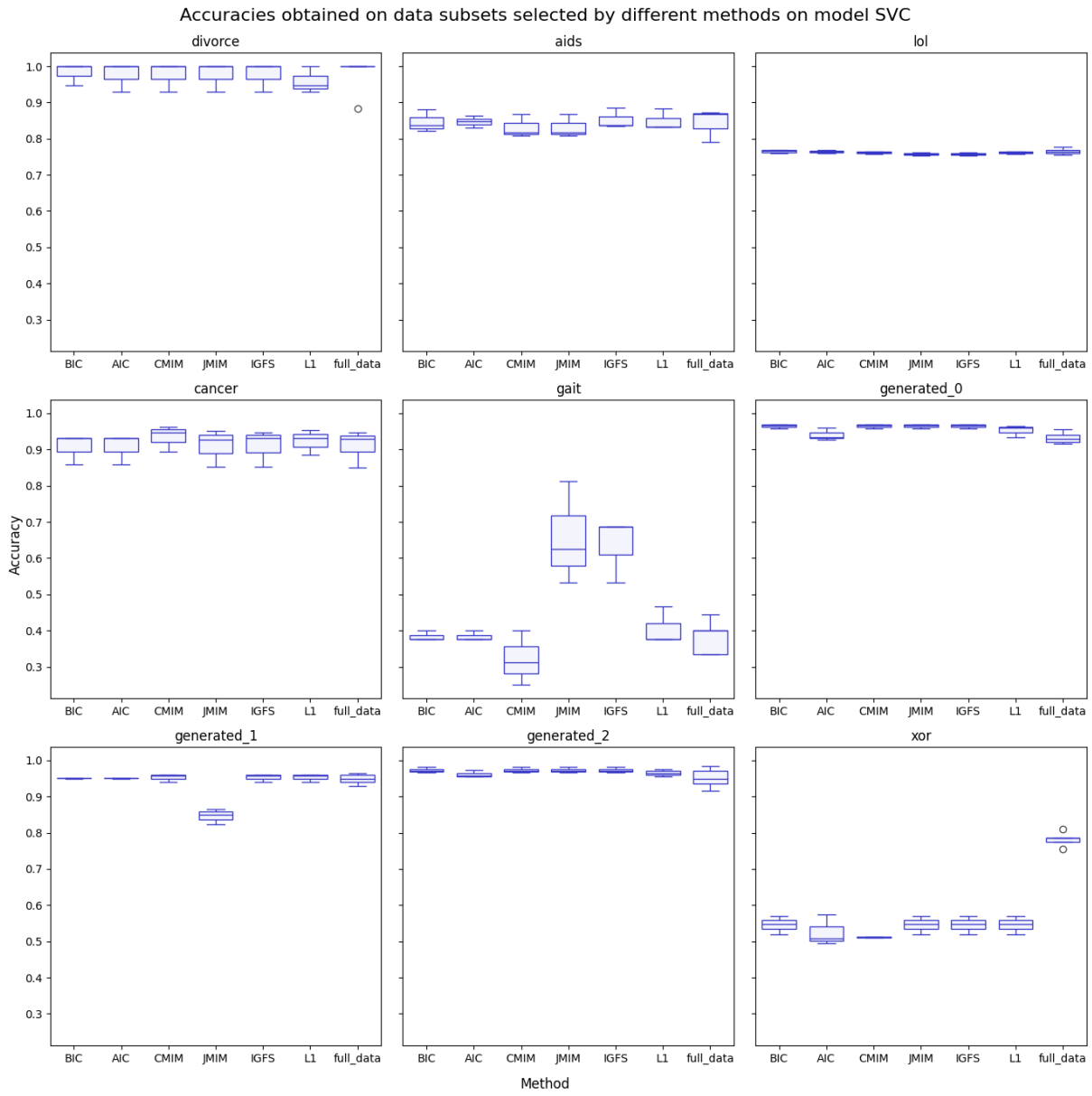

Figure 4: Accuracy scores for SVC models.

6 Conclusions

To sum up, we successfully implemented three Mutual Information-based feature selection methods and evaluated their performance across various datasets. For real-world datasets, models trained on features selected by the MI-based methods achieved slightly higher or very similar scores compared to other referential methods and training on the full dataset. On some sets MI methods chose very small subsets of features (e.g. one feature) and training on them got scores similar to the models on full data. For most of the generated datasets, the results were even more favorable, with MI methods accurately detecting relevant features with minimal errors. Among the MI methods, CMIM generally performed slightly worse than the other two methods. However, JMIM clearly underperformed on the `generated_1` dataset compared to other feature selection methods. The XOR dataset, designed to challenge MI-based methods, effectively fulfilled its purpose; moreover, other feature selection techniques also struggled with this dataset. For other three synthetic datasets our methods indicated the relevant features perfectly in almost all cases. Additionally, we successfully developed a stopping rule that performed well across all datasets. It is also worth noting that different models achieved varying results on the same sets of selected features. This indicates that even the best feature selector cannot compensate for an unsuitable model choice for a given dataset. Depending on the task and model, different feature selectors may perform best.

References

[1] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Networks 5*, pages 537–550, 1994.

[2] M. Bennasar, Y. Hicks, and R. Setchi. Feature selection using joint mutual information maximisation. *Expert Systems with Applications*, 42(22):8520–8532, 2015.

[3] A. El Akadi, A. Ouardighi, and D. Aboutajdine. A powerful feature selection approach based on mutual information. 8, 01 2008.

[4] F. Fleuret. Fast binary feature selection with conditional mutual information. *J. Mach. Learn. Res.*, 5:1531–1555, dec 2004.