Task for “A position for Software Developer in the field of Machine Learning and Explainable Predictive Process Mining”.

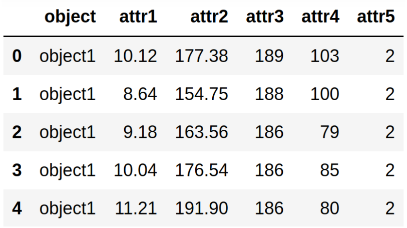
Alessandro Zuech

***Abstract*—This document describes the task completed for the application for “A position for Software Developer in the field of Machine Learning and Explainable Predictive Process Mining”. All the points of the task can be found in this document. Their description is placed after the related section title. An SVM and a MLP are trained on the same preprocessed dataset. Their hyper-parameters are fine-tuned to obtain the best evaluation metrics, which are used to compare the two classifiers.**

I. Preprocessing

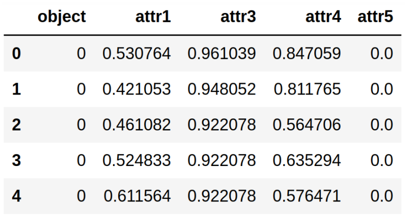
Task: preprocess the data into the required format for the model.

Data have been preprocessed from the original file (object.csv) into the required format for the model. The data consist of a table with 10000 rows and 6 columns, as in Figure 1. The column labeled ’object’ indicates the label, while the other columns show the attributes. The content of the first column is converted from a string to an integer by replacing object1 and object2 with 0 and 1. Then, each attribute column is normalized between 0 and 1. The result is shown in Figure 2.



	object	attr1	attr2	attr3	attr4	attr5
0	object1	10.12	177.38	189	103	2
1	object1	8.64	154.75	188	100	2
2	object1	9.18	163.56	186	79	2
3	object1	10.04	176.54	186	85	2
4	object1	11.21	191.90	186	80	2

Fig. 1. Content of the file containing the data.



	object	attr1	attr3	attr4	attr5
0	0	0.530764	0.961039	0.847059	0.0
1	0	0.421053	0.948052	0.811765	0.0
2	0	0.461082	0.922078	0.564706	0.0
3	0	0.524833	0.922078	0.635294	0.0
4	0	0.611564	0.922078	0.576471	0.0

Fig. 2. Data after normalization and converting the first column into binary value.

II. Statistical analysis

Data are analyzed to investigate their properties. The analysis of the samples shows that, out of a total of 10000 samples, each class has 5000 samples. The 2D scatter plots of the attributes show the correlation between the sample attributes for the object1 class (light blue) and the object2 class (orange). It can be observed that, for both classes, attr1 and attr2 are linearly dependent with each other (Figure 3). On the other hand, the other data features seem to be uncorrelated and show similar patterns. An example can be seen in Figure 4. From the 2D scatter plots the distribution of the data features resembles a gaussian distribution. It is also observable in a histogram plot of one of the features (Figure 5).

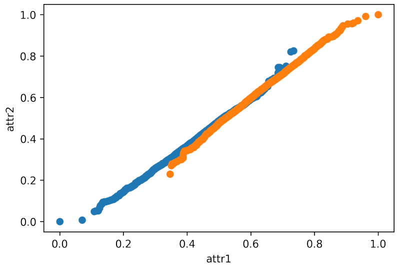


Fig. 3. 2D scatter plot of attr1 (horizontal axis) and attr2 (vertical axis).

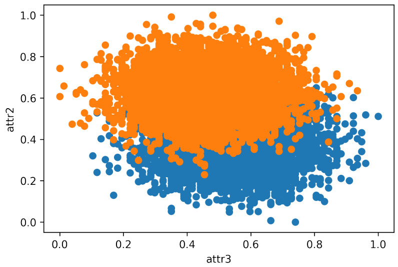


Fig. 4. 2D scatter plot of attr3 (horizontal axis) and attr2 (vertical axis).

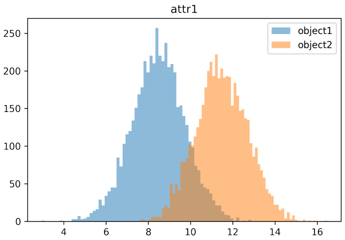


Fig. 5. Histogram plot of attr1.

III. Data splitting

Task: Split the data into train/test split based on the chosen split ratio (e.g., 80/20 train/test split, 70/10/20

train/validation/test).
Data are split using a ratio 80/20 train/test. By shuffling the dataset before splitting it into subsets, the samples of the two classes are uniformly distributed, therefore each subset contains the same number of samples belonging to the object1 and object2 classes.

IV. Machine learning classifier

Task: train one among the classical machine learning (non-deep learning) classifiers (e.g., decision tree, random forest, XGBoost, SVM) with the train dataset.
The machine learning classifier chosen for this task is the SVM. This machine learning classifier has been chosen to test its ability to correctly classify data that is not linearly separable. Since data appear not to be linearly separable, we can map them into a high-dimensional space with an adequate kernel function and then apply an SVM.

V. Artificial neural network classifier

Task: train an artificial neural network with the train dataset.
The artificial neural network model chosen for the binary classification task is a multi-layer perception (MLP) with all the hidden layers having the same number of neurons. This choice is meant to ease the hyper-parameter tuning, reducing the number of architectures to be tested. Also, since there are only 2 classes and 4 features, a highly complex architecture is probably not needed.

VI. Hyper-parameter tuning

Task: perform hyper-parameter tuning to improve the performance of the model.

A. Machine-learning classifier hyper-parameters

The machine learning classifier chosen for this task is a Support Vector Machine (SVM). The main hyper-parameter of the SVM is the kernel, used to map the observations into some feature space of higher dimension where they should be more easily separable, but each kernel contains some parameters itself. These are examples of basic kernels, where K represents the kernel function, x_i and x_j are training vectors and r and γ are hyper-parameters:

- **Linear kernel**: $K(x_i, x_j) = x_i^T x_j$. It is the simplest kernel for SVMs.
- **Polynomial kernel**: $K(x_i, x_j) = (r + \gamma \cdot x_i^T x_j)^d$. It is used to map the data in a higher dimension. For instance, with two features A and B, a polynomial of degree 2 would produce 6 features: 1 (any feature to power 0), A, B, A^2, B^2, and AB.
- **Radial kernel (RBF)**: $K(x_i, x_j) = e^{\gamma \cdot x_i^T x_j}$. This kernel is the most used and most successful kernel, due to the flexibility of separating observations with this method.

The linear kernel is not used, since the features are not linearly separable. The table in Figure I shows the values tested during the hyper-parameters tuning phase of the SVM. Note that the degrees hyper-parameter is ignored when the RBF kernel is used, and is set to -1 to avoid confusion.

TABLE I
Hyper-parameter values during tuning.

Parameter	**Values**
C	from 2^{-5} to 2^{16}
γ	from 2^{-15} to 2^{6}
Degrees	[2, 3, 4, 5]
Kernel	Polynomial or RBF

B. Artificial neural network hyper-parameters

The following list describes some hyper-parameters used in the MLP and how they have been set when testing the model.

- **Number of hidden layers**: they allow to model complex data distribution thanks to their neurons.
- **Number of neurons in the hidden layers**: they allow to combine information from the previous layer. When the number of neurons in the hidden layers is not high enough, the signal can not be adequately detected. On the other hand, using too many neurons can result in overfitting (the information processing capacity is much higher than the information in the training set) or increased training time.
- **Dropout rate**: the dropout rate is meant to avoid overfitting, and corresponds to the probability of setting the output of a neuron to zero. Generally, dropout can be set between 20% and 50%. A probability too low has minimal effect and a value too high results in under-learning by the network. If the network does not have a large number of neurons, a high dropout rate can decrease the network performance.
- **Learning rate**: the learning rate defines how quickly a network updates its parameters. A low value slows down the learning process, while a larger learning rate speeds up the learning but may not converge to a good local optimum. Usually, a decaying learning rate is preferred.
- **Momentum**: the momentum helps to know the direction of the next optimization step with the knowledge of the previous steps, helping to prevent oscillations.
- **Batch size**: the batch size is the number of samples given to the network during training. A good default for batch size might be 32.
- **Number of epochs**: the number of epochs is the number of times the whole training data is shown to the network while training. Ideally, the number of epochs should be increased until the validation accuracy starts decreasing even when training accuracy is increasing, indicating overfitting. This means that architectures with fewer parameters could perform better due to their faster optimization. For this task, the number of epochs is set to 200.
- **Loss type**: the loss type used for the MLP is the binary cross-entropy. This is usually used to measure the performance of binary classification problems.
- **Activation function**: activation functions are used to introduce nonlinearity to models, allowing them to learn

nonlinear prediction boundaries. Generally, the rectified linear unit (ReLU) is the most popular activation function. Sigmoid is used in the single node of the output layer when making binary predictions.

- **Network weight initialization**: since zero initialization is usually not successful in classification, uniform distribution is used.
- **Optimizer**: stochastic gradient descent (SGD) with momentum is used to optimize the MLP weights.

The table in Figure II shows the values tested during the hyper-parameters tuning phase of the MLP.

TABLE II
HYPER-PARAMETER VALUES DURING TUNING.

Parameter	**Values**
Hidden units	[2, 4, 6, 8]
Hidden layers	[1, 2, 3, 4]
Dropout rate	[0.1, 0.2, 0.3]
Batch size	[32, 64, 128]
Learning rate	[0.1, 0.01]
Momentum	[0.1, 0.2, 0.3]

VII. EVALUATION

Task: Evaluate both classifiers on the test dataset in terms of accuracy, precision, recall, F-measure, and AUC and compare their performances.

A. *Machine learning evaluation metrics*

Table III shows the best evaluation metrics obtained when testing a SVM. The parameters used to obtain those results are also reported in the table. In the column dedicated to the kernel type, Poly and RBF stand respectively for polynomial and radial basis functions.

TABLE III
BEST SVM SCORES AND RELATIVE HYPER-PARAMETERS.

Metric	**Score**	**Kernel**	**Degree**	γ	**C**
Accuracy	0.935	Poly	4	2^4	2^{-5}
Precision	1	RBF	-1	2^{-4}	2^9
Recall	1	RBF	-1	2	2^4
F-measure	0.934	RBF	-1	2^{-12}	2^{14}
AUC	0.935	Poly	4	2^4	2^{-5}

B. *Artificial neural network evaluation metrics*

Table IV shows, for each evaluation metric, the best obtained score and the hyper-parameters used for that neural network. Due to space constraints, the hyper-parameters in Table IV are referred with HU, HL, DR, BS, LR, M respectively for the number of hidden units, the number of hidden layers, the dropout rate, the batch size, learning rate and the momentum. The results show that a higher number of parameters (when 4 hidden units are used for each of the 2 hidden layers) is related to a slightly higher dropout rate, while the momentum is lower when the number of trainable parameters increases.

TABLE IV
BEST ANN SCORES AND RELATIVE HYPER-PARAMETERS.

Metric	**Score**	**HU**	**HL**	**DR**	**BS**	**LR**	**M**
Accuracy	0.931	2	1	0.1	64.0	0.1	0.3
Precision	1.0	2	1	0.1	128.0	0.01	0.2
Recall	1.0	2	1	0.1	32.0	0.01	0.1
F-measure	0.931	4	2	0.1	64.0	0.1	0.1
AUC	0.981	4	2	0.2	64.0	0.1	0.1

VIII. SUMMARY AND CONCLUSIONS

For this binary classification task, two classifiers have been chosen: a SVM and a MLP. They have been trained and tested on the same dataset of 4 normalized and independent features.

The similarity between the evaluation metrics shows that, for a binary classification task, a SVM is probably preferable to an MLP because of the SVM's lower training time.