Report on the analysis and modeling of the "steel-plates-fault" dataset

Introduction

The goal of the assignment was to analyze and model a selected dataset in order to expand one's knowledge related to data mining.

The repository includes a Jupyter nootebook containing the code and report, an analogous report in .pdf format, as well as .txt files and .png charts obtained during program execution.

Data description

A dataset named "steel-plates-faults" was used for the task. This dataset contains data on steel plate defects and consists of 1941 rows and 34 columns. Of these columns, 27 contain variables describing the physical parameters of the steel plate, 6 contain information on the occurrence of common defects (binary value 1-defect occurred, 0-defect did not occur, with only one of these columns being non-zero in a row). The last column, on the other hand, is the class of defect that occurred, which can take the value b'1' or b'2'. A value of b'1' means that one of the six most common defects occurred, while a value of b'2' means that another type of defect occurred. There are 1268 records of the first type and 673 of the second type in the collection.

In the .arff file provided for download, the column names are abbreviated and therefore unclear. For processing purposes, the column names have been changed in accordance with information shown on the source page.

Source:

Dataset provided by Semeion, Research Center of Sciences of Communication, Via Sersale 117, 00128, Rome, Italy.

https://www.openml.org/search?type=data&sort=runs&status=active&id=1504

http://archive.ics.uci.edu/ml/datasets/steel+plates+faults

Description of the process of preparing data for analysis and modeling

The dataset used is a dedicated dataset for use in machine learning. Therefore, it is complete and correct i.e. there are no missing values and no invalid values.

The data was used for the task proposed for this set, that is, classification into common defects and other defects.

Data analysis

First, the entire dataset was divided by strain class. Then Pearson's and Spearman's correlation coefficients were counted for each of them. Based on the results, it was decided to reject columns for which the correlation coefficient in both sets was greater than 0.85. 10 columns were rejected:

- TypeOfSteel_A400
- Y_Maximum
- X_Maximum
- Sum_of_Luminosity

- SigmoidOfAreas
- Orientation_Index
- Luminosity_Index
- LogOfAreas
- Log_X_Index
- Y_Perimeter

As a result, the dimension of the input vector used for modeling was reduced from 27 to 17 (28 to 18 including the expected value).

Then, based on the values obtained from the z-score function from the SciPy library, outlier records were rejected. Records for which the obtained value was greater than 3 were considered as such. There were 182 of them, which is about 9% of the entire dataset. After discarding the data, 1181 records of class b'1' and 578 of class b'2' remained.

An attempt was also made to use the Isolation Forest, but in that case about 14% of the dataset was rejected. This value was considered too high, so the previously mentioned method was used.

Data modeling

The columns that remained after the previous steps were used as the input vector X for the model. The values were previously normalized. The vector of expected values was then converted to a vector of binary values, 0 and 1.

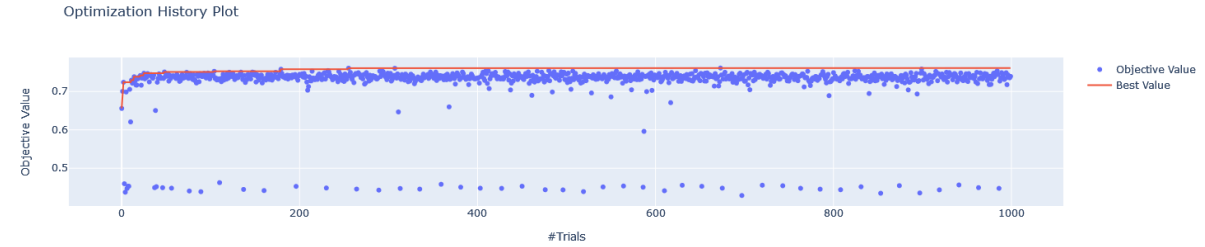
A support vector machine, specifically SVC from the scikit-learn library, was chosen to model the dataset. The optuna package was used to optimize the hyperparameters of the model. The value of the regularization parameter - C, the type of kernel of the support vector machine and the degree of the polynomial were optimized. At the same time, the degree of polynomial only affects the kernel of the "poly" or polynomial type, and is ignored in other cases. The maximized value is the sum of the accuracy on the training set, the precision on the training set, the f1 measure on the training set, the accuracy on the test set, the precision on the test set and the f1 measure on the test set. These values are summed with weights of 0.025, 0.025, 0.2, 0.075, 0.075 and 0.6, respectively. The average values of each metric for each trial were saved to a text file. At the same time, these values are actually averages obtained from 5 fold Cross-Validation.

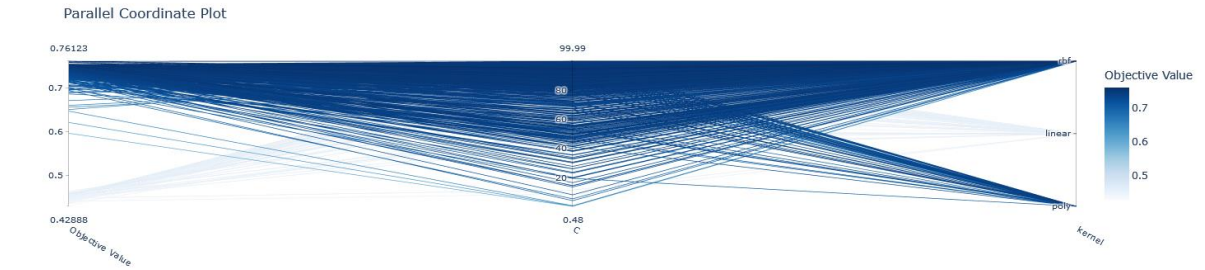
Optuna was used a total of three times, with 1,000 trials each time.

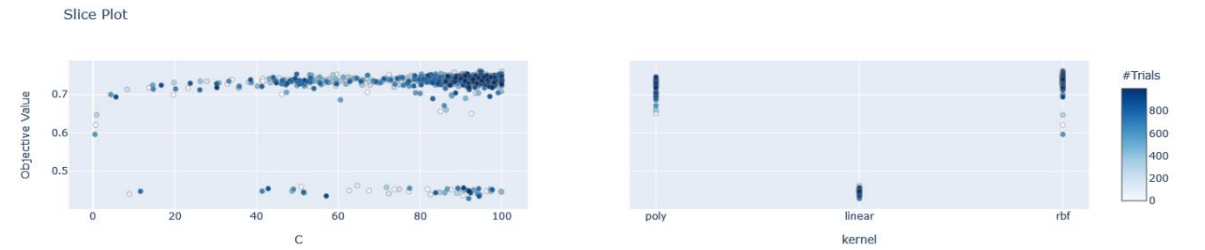
First, values were checked for C in the range 0.1 to 100.0, kernels of type 'linear', 'rbf' or 'poly', and the degree of the polynomial in the range 2 to 5. The best value was obtained for C = 94.54320781753651, kernel 'rbf' and degree = 5, where the degree does not matter because the kernel of type 'poly' is not used. The obtained values of the metrics for this example are shown in the table below.

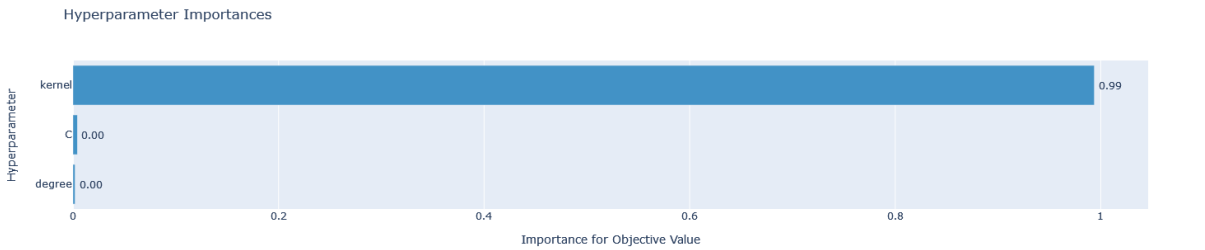
accuracy train	accuracy test	precision train	precision test	f1 score train	f1 score test
0.974986	0.778288	0.938107	0.613579	0.960972	0.644733

Additionally, visualization of the obtained results:





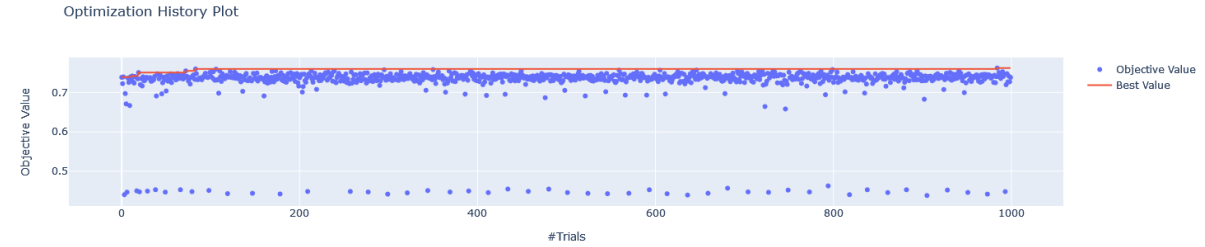


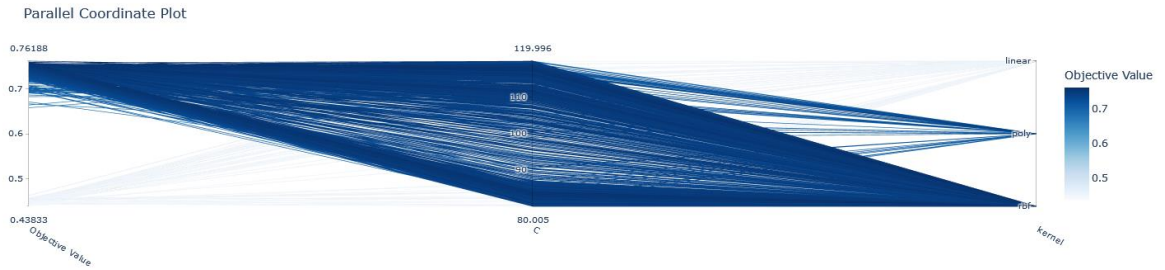


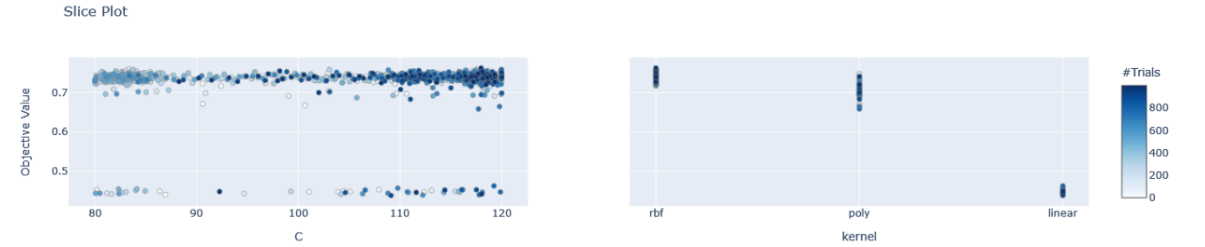
On the second attempt, based on previous results (Slice Plot), the search range of C was changed to 80.0 to 120.0, and the rest remained the same. The best value was obtained for C = 117.95849387058576, kernel 'rbf' and degree = 4, where degree does not matter because no poly kernel is used. The obtained values of the metrics for this example are shown in the table below.

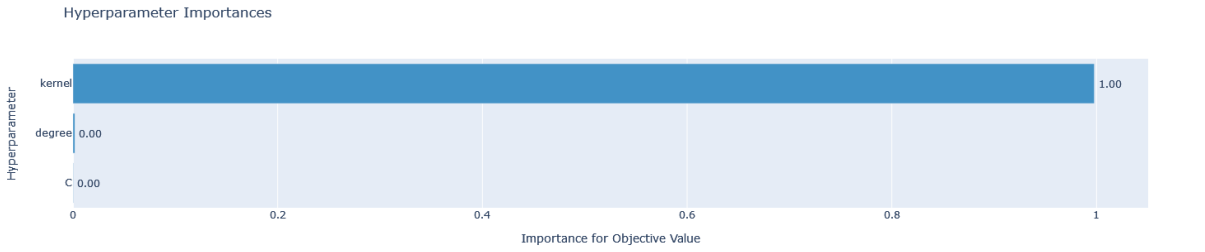
accuracy train	accuracy test	precision train	precision test	f1 score train	f1 score test
0.982377	0.793637	0.955512	0.67482	0.972732	0.68126

Visualization of the obtained results:





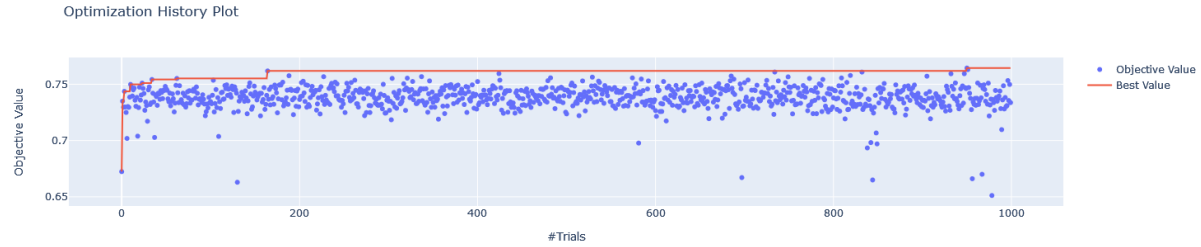


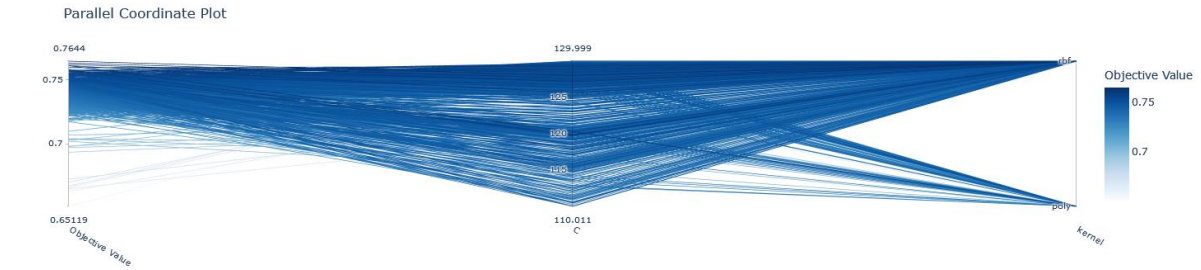


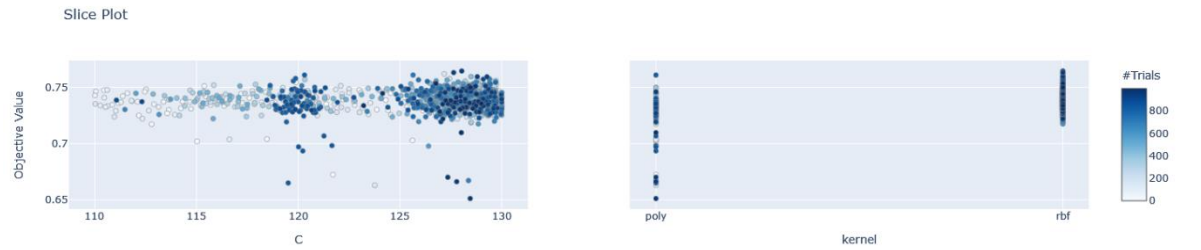
For the third attempt, the C-search range was changed to 110.0 to 130.0, and the linear kernel was no longer considered, due to the fact that it gave significantly weaker results than the other two. The rest of the settings remained as before. The best value was obtained for C = 128.0280732226804, kernel 'rbf' and degree = 3, where degree does not matter because the "poly" type kernel is not used. The obtained values of the metrics for this example are shown in the table below.

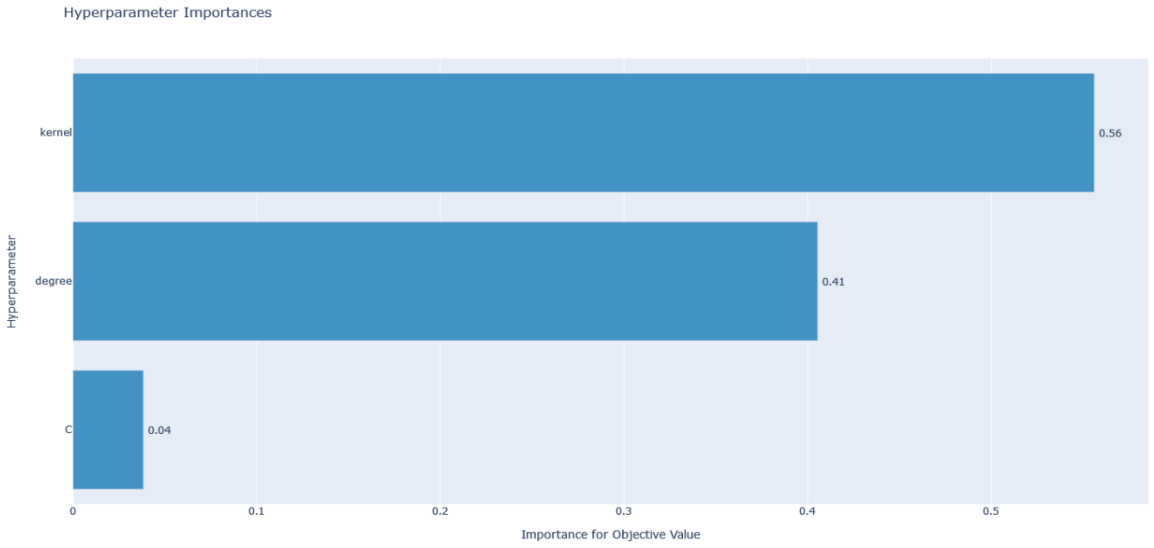
accuracy train	accuracy test	precision train	precision test	f1 score train	f1 score test
0.98266	0.79363	0.95844	0.683672	0.973181	0.68407

Visualization of the obtained results:









Due to the small improvement in parameters, the result obtained was considered the final result.

Conclusions

In the end, the best model was obtained for C = 128.0280732226804, kernel 'rbf' and degree = 3, where degree does not matter because the poly kernel is not used. The obtained values of the metrics are as follows:

accuracy train	accuracy test	precision train	precision test	f1 score train	f1 score test
0.98266	0.79363	0.95844	0.683672	0.973181	0.68407

Because the project is essentially self-developing, it is hard to determine whether the result obtained is satisfactory and whether discarding columns with correlated values was a good move. In addition to the result obtained, the project demonstrates how much the hyperparameters of the model can impact the final outcome. An unusual error occurred during the project's implementation. Specifically, when running the project with JupyterLab in the Firefox browser, calling "optimize" on the "study" object from the optuna package caused the browser to progressively consume more RAM. Before completing 1,000 trials, the browser exceeded the available RAM on the machine, leading the operating system to shut it down. Eventually, this issue was resolved by using the Microsoft Edge browser.