Inteligencia Artificial Explicable

Hands-on Project Unit 2

Submission

- You may carry the project out in groups of two students.
- Must upload the deliverables to Moodle before the deadline. **Note: A single submission per group**.

Deliverables

1. A PDF file with a project report (maximum 15 pages). Must contain the names of group members.
2. All code used to carry out the analysis:
    1. Python: A notebook that must run, out of the box, on Google Colab.
    2. R: Script or RMarkdown file that must run, out of the box, in a typical R installation (e.g., please include code for installing required packages).

Phases

The project will be delivered in **two phases**. Regarding the first phase:

- use only the decision tree;
- corresponds to 10% of the final grade (no minimum grade).
- deadline: 29/09/2024 at midnight.

The second phase corresponds to the delivery of the final and complete report and corresponds to 90% of the final grade.

What to do?

Mandatory

You will train and interpret glass-box models. In particular, you will train at least two of the following classifiers:

- Logical (trees/lists/sets);
- Linear;
- Generalized additive models (Explainable boosting machine).

You may use any learning algorithm for the selected models, including standard greedy decision tree methods such as CART. You will also need to evaluate each of the classifiers on validation data. You will need to carry out both global and local interpretation; that is, interpret the learned models as well some of its predictions. In particular, regarding local explanations I recommend selecting 2-4 instances (at least one of each class) and explain their predictions with different models.

Optional

The following are examples of additional tasks that can increase your grade:

- Try to increase the performance of the classifier on validation data, while using *training data only* to fit them – using validation data for training will lead to heavy penalties. This includes parameter tuning, preprocessing, etc.
- Try non-standard learning algorithms, such as those for learning optimized trees, or methods that we have not covered in class, such as RuleFit. Before using a method that we have not covered, check with the professor that the model is adequate.
- Evaluate black-box models such as random forest or gradient boosting, on the validation set, and compare to the performance of glass-box models.
- Discuss differences in performance among the different models.
- Critical discussion of the process and/or obtained results.
- ...

The above list is not exhaustive and you may try to improve the grade in a different way. If in doubt, check with the professor.

Note: some of these tasks may be time-consuming and yet produce no tangible results. For example, you might not be able to improve over the baseline method with parameter tuning, or the selected algorithm/package may be too slow. Before investing significant into into such a tasks try to improve the project in 'more straightforward' ways.

Data

The training set is contained in the **train.csv** file and the validation set in the **validation.csv** file. Both files have eight columns, corresponding to eight variables. These are all explained in the data dictionary file, .dictionary.xlsx.

The first variable, 'RiskPerformance', is the target variable, to be predicted from the rest. The monotonicity constraints are absolutely optional and, of course, cannot be enforced with out-of-the-box models.

Code

You are free to use any Python or R package. You may combine Python and R. Some suggestions:

- scikit-learn;
- interpretml (https://interpret.ml/docs/dt.html);
- imodels (https://github.com/csinva/);
- CORELS (https://corels.cs.ubc.ca/corels/).

Report structure

The report may be written in either Spanish or English. The structure of the report is as follows:

1. Cover page. Include a cover page with title, authors' names, email, course, and date. **Make sure to list all authors (group members)**.
2. Introduction. Briefly explain the problem, the data, your overall approach and give overview of your results.
3. Method. Indicate your strategy and reasoning for fitting of each of the classifiers, justifying the choice of the algorithm, parameters and configurations. You may discuss why certain options might make sense and while others might not.
4. Results. Provide validation set accuracy results for each classifier.
5. Models. Interpret the learned models. This is the most important section.
6. Discussion: Discuss the models' performance and interpretability (subjective). Could some aspects be improved? For example, if you converted a tree to a rules list, could the list have been further simplified?

7. Conclusion. Provide conclusions regarding the project, your approach and your results. Discuss possible additional steps towards improvement.
8. References.

Results summary

Important: The Results section must begin with a table summarizing considered methods and their best prediction results. There will need be a row for each model, and a column for the best non-default parameter settings for that class as well as for the best training set and validation-set accuracy obtained with those settings. For example:

Classifier	Best parameters	Train accuracy	Test accuracy
ExplainableBoostingMachine		0.85	0.75
RandomForestClassifier	n_estimators=100	0.85	0.75
...	...	...	...

Grading

Achieving a grade of 5 requires:

- Delivering all mandatory content, as specified above;
- Not-below-chance accuracy with each model;
- Absence of major conceptual or formal mistakes.

Secondary criteria include:

- The quality and clarity of the explanations, justifications, the writing and the presentation. Balancing clarity with rigor.
- Use of plots and other aids that help understanding the report.
- Extent of originality. Use of own data processing, methods, analysis.

Important: There will be penalties for not following submission instructions, such as exceeding the page limit, not providing the code, etc. Additional recommendations:

- Not everything that you do needs to appear in the report; keep the report concise and to the point. I can check the notebook to see all necessary details.
- Avoid plain copy-paste of code and output in the report; present the results in a clear and concise way. In particular, **do not deliver Jupyter or RMarkdown notebooks rendered as pdf**.
- Watch out with figures: for example, ensure the axis' scales are discernible.
- I encourage discussion of the achieved results, the motivation for trying some options and not others, and the reasoning behind the choices made, etc.