Information Visualization (Fall, 2022)

Project 1

Project 1

Summary

In this project, you will have to create the simples and clear visualisation for a given dataset. You have to follow the best practices taught in the class to use proper chart types and implement them in a way that is easy to read and understand by a general audience.

Submission

- Deadline: September 30.
- Group work in teams of **4 students**.
- The submission file needs to have the following file name format: `group_<group_id>_<student1_id>_<student2_id>_<student3_id>.zip`.
- (Alternatively, you can simply submit the link to a Github repository.)

Dataset

In this project you will have to work with an existing dataset, available here. The dataset was collected from a study that compares the energy consumption of training a machine learning model across different training algorithms and different dataset properties (i.e., number of features, number of data points, and type of data).

It contains 17 features:

- `algorithm` the machine learning algorithm (SVM, Decision Tree, Multinomial NB, KNN, Random Forest, AdaBoost, Bagging Classifier)
- `RQ`. Research question being study.
  - `RQ2.1` refers to what is the impact of the size of the training data set (`no_datapoints`) in `train_energy(J)`.
  - `RQ2.2` refers to the impact of the number of the features in the training dataset (`no_features`) in `train_energy(J)`.
  - all these research questions were tested acroos all the different machine learning algorithms.
- `experiment_id`. Unique identifier of the experiment.
- `iteration`. Repetition number. Each experiment is repeated 5 times.
- `no_datapoints`. Number of rows in the dataset used to train the machine learning model.
- `no_features`. Number of features in the dataset used to train the machine learning model.
- `preprocessing_energy(J)`. Energy consumption of the preprocessing stage of the machine learning model.
- `preprocessing_time(s)`. Duration of the preprocessing stage of the machine learning model.
- `train_energy(J)`. Energy consumption of training the machine learning model.
- `train_time(s)`. Duration of training the machine learning model.
- `predict_energy(J)`. Energy consumption of making predictions in the machine learning model.
- `predict_time(s)`. Duration of making predictions in the machine learning model.
- `datatype`. Data type used to store the training dataset
- `accuracy`. Accuracy achieved by the machine learning model.
- `precision`. Precision achieved by the machine learning model.
- `recall`. Recall achieved by the machine learning model.
- `f1`. F1-score achieved by the machine learning model.

Although the dataset has 17 features. For this project, you will focus on 4 main features. 3 independent variables: `no_datapoints`, `no_features`, and `algorithm`. 1 dependent variable: `train_energy(J)`. In sum, we want to make visualizations that show how the independent variables affect energy consumption — i.e., the dependant variable.

Requirements

1- Perform an exploratory analysis of the dataset. Remember that exploratory analyses do not need to be refined or clear. It is just a draft of several visualizations that help get familiar with the data.

- Hence, in this step you should have several visualizations that help getting an idea of the datasets and will serve as a starting point to the **explanatory visualizations** in this project. (Don't forget the difference between exploratory and explanatory).
- **Hint:** Python notebooks are usually useful for exploratory analyses because you can combine python code, visualizations and markdown text. Use all of these elements.

2- Create a visualization that shows, for the algorithm `SVM`, how `no_features` affects energy consumption (`train_energy(J)`). This visualization should 1) choose the most suitable chart type, 2) follow the visualization guidelines taught in the class, and 3) be as simple as possible. If necessary, there should be a visual element highlighting how `no_features` correlates with `train_energy(J)`.

- Note that each experiment is repeated 30 times (as denoted by the feature `iteration`). You may want to use the average of these 30-sized samples and maybe its standard deviation.

3- As you can imagine, creating a simple visualization to show all results is far from trivial. There are 3 main variables that are compared against `train_energy(J)`: `no_datapoints`, `no_features`, and `algorithm`. Create a visualization using small multiples that shows, **for each machine learning algorithm**, how `no_datapoints` and `no_features` affect energy consumption (`train_energy(J)`).

4- Create a single plot that is able to capture most of the insights of the visualization in requirement 3. You wont be able to capture all the insights, but the idea is the capture as much as possible while **keeping the visualization simple and interesting**.

5- Create a visualization that shows an interesting insight in the data that was not unveiled by the visualizations of requirements 2, 3, and 4.

Important Notes

- Requirements 2, 3, 4 should be modular. I.e., it should be easy to reutilize the graph in a different project. To achieve it, you can for example implement part of the graph as a generic method that receives data (the same way we did with the skinny plots of assignment 6).
- Use the potential of Jupyter Notebooks to make the submission appealing and easy to read. For example, use the markdown cells to structure the notebook.
- To import the dataset you can use Pandas or the standard libraries of Python.

Grading

This project affects 35% of the final grade. The following rubric items will be considered:

- Overall cohesiveness of the project.
  - To what extent is the team work cohesive?
  - Does the report completely describe the plots and how to read them?
- Quality of the submission.
  - Does the code follow code conventions?
  - Does the code execute without any issues?
- Clarity
  - Do visualizations follow best practices?
  - Are visualizations easy to read?
  - Did the students make an effort to present data in simple way?
- Relevance.
  - Are the visualizations showing relevant patterns of the data.
- Creativity.
  - Did the authors any unusual, yet interesting visualization?
- Robustness.
  - Does the plot work under different testing settings?
- Customization.
  - Is it possible to customize different aspects of the plot? E.g., color, transparency, etc.
- Organization.
  - Are the submission artifacts sound and clear?
  - Is the report easy to read?

Submission status

Group	Group24
Submission status	Nothing has been submitted for this assignment
Grading status	Not graded
Due date	Friday, 16 September 2022, 12:00 AM
Time remaining	2 days 8 hours