École national Supérieure d'Informatique (ESI), Algiers, Algeria
2CSSID (2025/2026)
Machine Learning (ML)

Workshop 02: Data preparation and models' evaluation

Mr. Abdelkrime ARIES

Apply some data preparation techniques
Get familiar with CRISP-DM on a simple project
Learn models' evaluation
Try some tools: pandas, matplotlib, scikit-learn, imbalanced-learn

Useful links

- **Dataset**: `https://archive.ics.uci.edu/dataset/601/ai4i+2020+predictive+maintenance+dataset`
- **Demo**: `https://github.com/projeduc/ESI_ML/blob/main/TP/SD_2022-2023/TP01/TP01_pretraitement_sujet.ipynb`
- **MatplotLib**: `https://matplotlib.org/stable/gallery/index.html`
- **imbalanced-learn**: `https://imbalanced-learn.org/stable/user_guide.html#user-guide`
- **CRISP-DM (web)**: `https://www.ibm.com/docs/en/spss-modeler/18.5.0?topic=dm-crisp-help-overview`
- **CRISP-DM (pdf)**: `https://www.ibm.com/docs/en/SS3RA7_18.5.0/pdf/ModelerCRISPDM.pdf`

1 Business understanding

The first step is to get a clear understanding of the business goals and needs for this project. Then, we will translate this knowledge into a well-defined data mining problem. Finally, we will create a preliminary plan outlining the steps needed to achieve those goals.

Predictive Maintenance is a proactive approach to equipment maintenance. It uses data analytics and machine learning techniques to predict when machinery or systems are likely to fail. In our case, not only we want to detect if a machine is going to fail; but to detect one or more of these 5 failure modes: tool wear failure (TWF), heat dissipation failure (HDF), power failure (PWF), overstrain failure (OSF) and random failures (RNF).

To this end, we want to use all or part of these features:

- product ID: consisting of a letter L, M, or H for low (50% of all products), medium (30%) and high (20%) as product quality variants and a variant-specific serial number
- air temperature
- process temperature
- rotational speed: frequency of rotation of an object around an axis.
- torque: a measure of the force that can cause an object to rotate about an axis.
- tool wear: gradual failure of cutting tools due to regular operation

Discuss these solutions, given we want to use logistic regression:

1. One model having the 5 outputs (TWF, HDF, PWF, OSF, RNF) with softmax function.
2. One model having the 5 outputs (TWF, HDF, PWF, OSF, RNF, Other) with softmax function.
3. One model having the 5 outputs (TWF, HDF, PWF, OSF, RNF) with sigmoid function.
4. One model having the 5 outputs (TWF, HDF, PWF, OSF, RNF, Other) with sigmoid function.
5. Two models, one for binary classification for failure yes/no. In case of failure, we train a second model to detect the type similar to that of first solution.
6. the same thing, but the second model is similar to that of the third solution.

2 Data understanding

In the data understanding phase, we first gather the available information. Then, we dive deep to get to know the data: We check for quality issues, look for early clues about trends, and identify any intriguing patterns. This lets us form hunches about what the data might be hiding.

2.1 Collecting initial data

Let us suppose that data exists already. But, we have multiple sources: an sqlite3 database and a csv tabular.

- Read the data using **pandas**:
  - Read **ai4i2020a.csv** which is a CSV file with comma as separation mark
  - Read **ai4i2020b.sqlite** which has the following table

```
CREATE TABLE failure (
    ID VARCHAR(7) NOT NULL,
    Air_Temperature_K FLOAT,
    ProcessTemperature_K FLOAT,
    Rotational_speed INTEGER,
    Torque FLOAT,
    Tool_wear INTEGER,
    Failure INTEGER,
    TWF INTEGER,
    HDF INTEGER,
    PWF INTEGER,
    OSF INTEGER,
    RNF INTEGER
);
```

2.2 Describing data

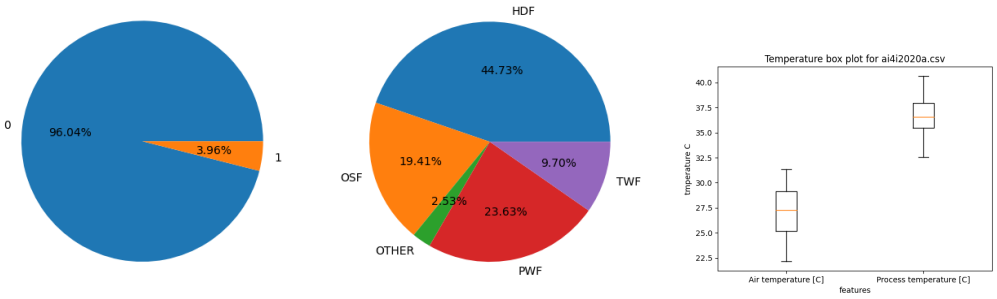
For each dataset, apply these techniques:

- Show columns info using **pandas.DataFrame.info**().
- If numerical values are not detected properly, fix it (check data types on the online dataset's description).
- Describe the dataset using **pandas.DataFrame.describe**().
- Describe dataset 1 (CSV) when "*Type*" == "L" (low).
- Show categories of non-unique categorical features.

2.3 Exploring data

For each dataset, apply these techniques:

- Plot a pie chart for binary relation failure/not (Fig. 1(a))
- Plot a pie chart for different failure classes (Fig. 1(b))
- Draw the box plot of to temperature features for non null values (Fig. 1(c)).
- If you detect any outliers, display them.
- Draw plot pie chart of "*Type*".
- Apply PCA on "*Air temperature*", "*Process temperature*", "*Rotational speed*", "*Torque*" and "*Tool wear*" to get a two dimensional representation.
- Plot samples' distribution based on this representation so we can check classes' separability



(a) Binary relations (b) % classes (c) Box plot

Figure 1: Exploring results

2.4 Verifying data quality

- Show samples containing at least one omitted value
- Show samples which do not have a failure, but have a failure type (contradiction)

3 Data preparation

3.1 Selecting data

- Do we need these features: "*Product ID*" and "*Machine failure*"? Why?

3.2 Constructing new data

- Transform Celsius into Kelvin for temperature : $K = C + 273.15$
- Transform Failure types into one categorical column and delete original columns (dataset 1)
- Add another column for the failure type "OTHER" (dataset 2)
- If "*Type*" does not exist, infer it from "*Product ID*"
- Transform "*Type*" into OneHot representation

3.3 Integrating data

- Rename columns to get the same schema.
- Delete unshared columns (which exists only in one dataset).
- Merge the two tables into one.

3.4 Cleaning data

- Delete all inconsistent samples (there is no failure yet there is a failure type)
- Find all duplicate samples using "*product ID*"
- Replace all missing values from the same product (using "*product ID*")
- Delete duplicates
- Delete "*Product ID*" and "*Failure*" (no need for those anymore)
- If a sample has more than a missing value, delete it
- Replace missing values by the average of the same type

3.5 Formatting data and Constructing new data

- Put the classes at the end so we can select them easily.
- Split data into train (70%) and test (30%).
- Apply the oversampling **SMOTE** on training dataset to get a new one
- Apply two undersampling techniques on the majority class: **ClusterCentroids** and **TomekLinks** to get two new training datasets
- Apply normalization on the 4 past datasets to get three new ones
- In this case, we will have 8 training datasets: No_sampling, No_sampling_normalized, SMOTE, SMOTE_normalized, ClusterCentroids, ClusterCentroids_normalized, TomekLinks and TomekLinks_normalized.
- Separate inputs (X) and outputs (Y) for each of them
- Transform Y's into OneHot preserving the original one

4 Modeling and Evaluation

- Train these models on the 8 datasets:
  - Logistic regression with L2 penalization (by default) and "liblinear" solver.
  - Gaussian Naïve Bayes
  - CART Decision tree
  - Random forest with 50 estimators
- For each of the 32 resulted models, get these information:
  - Train time
  - Test time
  - Train accuracy
  - Test accuracy
- Show the table which summarizes these results

Let's discuss these results:

- In terms of training time
  - What is the order of the algorithms from fastest to slowest?
  - Justify this based on how the algorithm constructs the model.
  - How can we improve the training time of each algorithm (if we can)?
- In terms of test time
  - What is the order of the algorithms from fastest to slowest?
  - Justify this based on how the model estimates the class.
  - How can we improve the estimation time of each algorithm (if we can)?
- In terms of training accuracy
  - What is the order of the algorithms from best converged to worst?
  - Justify this based on how the algorithm constructs the model.
- In terms of test accuracy
  - What is the order of the algorithms from best generalized to worst?
  - Justify this based on how the model estimates the class.
- In terms of used data
  - How normalization has affected the results?
  - How sampling (over-sampling and under-sampling) has affected the results?