Computer Engineering Project

Data Mining of Scientific Data

POLITECNICO
MILANO 1863

Giulio Occhipinti

Supervisor: Barbara Pernici
Co-supervisor: Edoardo Ramalli

A.A. 2021-2022

Contents

1 Introduction

A *model* has the functionality of predicting one or multiple properties of an experiment given a certain input. Each experiment can be simulated with a model to derive the simulated data. One of the most important steps in the model development procedure is model validation, in which the experimental data is compared against the simulated one, therefore, it is possible to quantify the similarity between simulated data and the experimental data. In our scenario, the similarity function used is called *curve matching*: it can be used to tell whether a model is more or less accurate at simulating a certain experiment by looking at its output value, the *score*, which is the average of five indices: d0L2, d1L2, d0Pe and d1Pe. These are *dissimilarity indices* which measure different aspects of the curves. A score of 1 indicates perfect matching, while 0 is the opposite[1]. Each experiment has a number of categorical (experiment type, reactor, target, fuels) and numerical properties (pressure, temperature and phi ranges). A *reactor* is the experimental facility where the experiment takes place. The *target* is the property that is being observed during the experiment, usually the concentration of a substance.

This project has two main objectives:

1. Comparing two models to find out which model is more or less accurate at a certain combination of metadata, (or more precisely, a certain permutation of experiment type, reactor, target and fuels used). Given two models, it may be more appropriate to use one instead of the other.

2. Finding out in which conditions and why a certain model is performing poorly at a certain experiment type compared to the other ones (*why is this model's average score for this permutation so much lower than the other averages?*)

Machine learning algorithms and data mining libraries in Python such as Pandas and scikit-learn have been used to analyze the models and find the answer to these two questions. In the following sections I'm going to introduce the concepts of correlation and clustering, both of which are used in the project: the correlation was used to find out about any relationship between the score and another variable of the model, while clustering was used to have a more accurate score average when a permutation has a particularly elevated amount of experiments. I will explain how I used them and what their purpose is in Section 2.

[1]Ramalli, Edoardo. "Data Ecosystems for Scientific Experiments: Managing Combustion Experiments and Simulation Analyses in Chemical Engineering" (September 2021): 6-7. https://doi.org/10.3389/fdata.2021.663410

1.1 Correlation

Two variables (in the context of this project, the variables are the metadata of each experiment, such as the score, fuels, and temperature) are *correlated* when there is a relationship between the values of one variable with the other. For example, let X and Y be two numerical variables: if the value of Y increases when the value of X also increases, then there is most definitely a positive correlation between X and Y. To give a more practical example, let X be the age of a person and Y their height. When X ranges between 0 and 18, there's a positive correlation between the two variables. If one variable decreases while the other one increases, then they have a *negative correlation*. In order to quantify the correlation between two variables, it is necessary to find a *correlation coefficient* which tells us how strong the correlation between them is. Depending on the type of variable (numerical or categorical), there are different ways to calculate the correlation coefficient.

Definition 1.1 (Numerical Variable)**.** A numerical variable is a variable whose values are numbers that represent a quantifiable characteristic, such as a measurement or the result of a calculation. Let x be the value of a numerical variable. x can be continuous ($x \in R$), or discrete ($x \in N$). For example, a continuous numerical variable could be a measurement of a height or a temperature, while a discrete variable could represent an age.

Definition 1.2 (Categorical Variable)**.** A categorical variable is a variable that only takes values from a limited set, for example sex (male or female) or a rating (good, bad or neutral). A categorical variable can take numerical values, but that does not make it a numerical variable: for example, (good, bad, neutral) can be written as (0, 1, 2) but it is still a categorical variable.

1.1.1 Two numerical variables

In this section we are going to see three different correlation coefficients: Pearson's, Spearman's and Kendall's. I am going to explain the differences between them, and when each coefficient should be used depending on the characteristics of the numerical variables.

Pearson Pearson's coefficient is used to find a linear correlation between two variables. Let X and Y be two numerical variables.

Pearson's coefficient should be used if:

- It makes sense to compare the two variables using their numerical values, for example if X and Y are not rankings but rather measurements.
- The two variables can have a linear correlation (if by plotting X and Y there clearly isn't a linear correlation, it would be more appropriate to use another coefficient).
- The two variables are normally distributed, meaning that both X and Y have a Gaussian distribution.

The coefficient is calculated with the following formula:

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y}$$

Where cov(X, Y) is the covariance between X and Y and σ_X and σ_Y are X and Y's standard deviation.

- If $\rho = \pm 1$ the variables are completely correlated: if X increases, Y increases ($\rho = 1$) or decreases ($\rho = -1$).
- If $0 < \rho < 1$ (or $-1 < \rho < 0$) the variables are more or less correlated.
- If $\rho = 0$, the variables are completely unrelated.

It is possible to use Pearson's coefficient to find the correlation between two or more variables (independent variables) and another one (dependent variable). This multiple correlation coefficient is referred to as **R**.

$$R = \sqrt{C^T M^{-1} C}$$

Where $C = [\rho_{Y,X}, \rho_{Z,X}]$ and M $= \begin{pmatrix} \rho_{Y,Y} & \rho_{Y,Z} \\ \rho_{X,Y} & \rho_{X,Z} \end{pmatrix}$ is the correlation matrix

Spearman Spearman's coefficient is used to find a monotonic relation between two variables. It is used if at least one requirement to use Pearson's is not met. Spearman's coefficient is Pearson's coefficient between the rank of the two variables, rather than their numerical value:

$$\rho_s = \frac{cov(R(X), R(Y)}{\sigma_{R(X)} \sigma_{R(Y)}}$$

Where R(X) and R(Y) are the ranks of the two variables. If all n ranks are distinct integers, meaning there are no equal pairs in the table that has X and Y as columns (X=x, Y=y), then it is possible to use the following simplified formula:

$$\rho_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Where n is the number of observations, and $d_i = R(X_i) - R(Y_i)$ is the difference between the ranks of each pair (x, y).

Kendall Kendall's coefficient is very similar to Spearman's because it is also used to find how strong is the monotonic correlation between two numerical variables based on the ranks of their values. The difference is that Spearman's coefficient requires less calculations, and is therefore easier to process[2]. However, Kendall's is more robust and accurate compared to Spearman's because its distribution approaches normality more rapidly.

[2]Spearman's coefficient has $O(nlog(n))$ time complexity, while Kendall's is $O(n^2)$

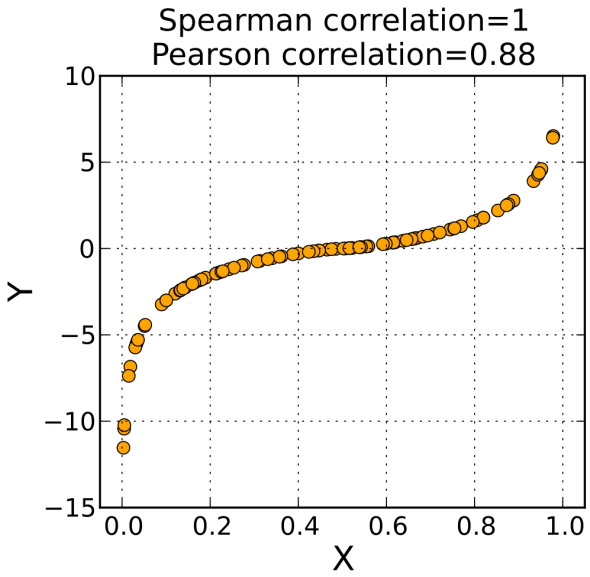

Figure 1: Unlike Pearson's coefficient the correlation is not linear, but rather monotonous: that means that with a high Spearman's coefficient Y increases as X increases, but not necessarily by a constant amount.

$$\tau = \frac{(C - D)}{(C + D)}$$

Where C is the number of concordant pairs and D is the number of discordant pairs.

1.1.2 Two categorical variables

Cramer Cramer's V coefficient is an extension of the ϕ coefficient which makes it possible to find the correlation between categorical variables with more than two possible values (e.g. yes/no), which is why it can also be referred to as ϕ_C. The coefficient can take values between 0 (no correlation between the variables) and 1 (the variables are completely correlated). Like nearly all methods to find the correlation between categorical variables, the calculation of Cramer's coefficient is based on the contingency tables, tables where each cell contains the number of instances of every couple of values (X = x, Y = y). For example, if X = {man, woman} and Y = {yes, no}, the contingency table will look like the one pictured in figure 2.

		Gender	
		Male	Female
Grant	Yes	100	40
	No	600	500

Figure 2: A contingency table. In the original table with columns [Gender, Grant] there are 100 (male, yes) pairs, 40 (female, yes) pairs, and so on.

Cramer's V is calculated with the following formula:

$$V = \sqrt{\frac{X^2}{n * min(k-1, r-1)}}$$

Where n is the total number of observations (the number of rows of the input table), k and r are respectively the number of columns and rows of the contingency table (the number of possible value that X and Y can take), and X^2 is Pearson's chi-squared test:

$$X^2 = \sum_{i=1}^{n} \frac{(O_i - E_i)^2}{E_i}$$

Where:

- n is the number of cells of the contingency table
- O_i is the number of observed cases of the (x_i, y_i)-th pair (the value of the cell in the contingency table)
- E_i is the expected value of every cell in the contingency table, which can be calculated by doing $\frac{(totalRow * totalColumn)}{totalTable}$, where $totalRow$ and $totalColumn$ are the sum of the values of the cells in the contingency table that have the same row or column as the i-th cell.

1.1.3 One numerical and one categorical variable

Correlation ratio The correlation ratio measures the relation between the variance of the values in each category and the variance of the values as a whole (the numerical variable). In other words, it tells how accurately whether a value of the numerical variable belongs to one of the categories of the categorical variable. The correlation ratio ranges from 0 to 1, where a value close to 0 means there is no dispersion between the means of the different categories, while a value close to 1 indicates that there is no dispersion within the categories themselves. In the rare case that the data has the same values in every category, the correlation ratio is undefined.

The correlation ratio is calculated with as follows:

$$\eta^2 = \frac{{\sigma_{\overline{y}}}^2}{{\sigma_y}^2}$$

Where:

- $\sigma_{\overline{y}}^2 = \frac{\sum_x n_x (\overline{y}_x \overline{y})^2}{\sum_x n_x}$ is the weighted variance of the category means
- $\sigma_y^2 = \frac{\sum_{x,i} (y_{xi} \overline{y})^2}{n}$ is the variance of all samples

1.2 Clustering

Clustering is a machine learning algorithm that divides the input variables (in this case the rows of the model) in different clusters. In order to have an easier understanding of how the data is clustered, it is recommended to show a scatterplot where each cluster has a different color, and each point's parameter can be viewed easily. Clustering can be useful to study the behavior of data by seeing what the points belonging to a certain cluster have in common, or how the clusters are positioned in the plane (2D) or space (3D). There are different ways to cluster a dataset: affinity-propagation, DBSCAN, OPTICS, and many others. I decided to use K-means because of how effective it is, despite being relatively simple to implement and not expensive resource-wise compared to other algorithms.

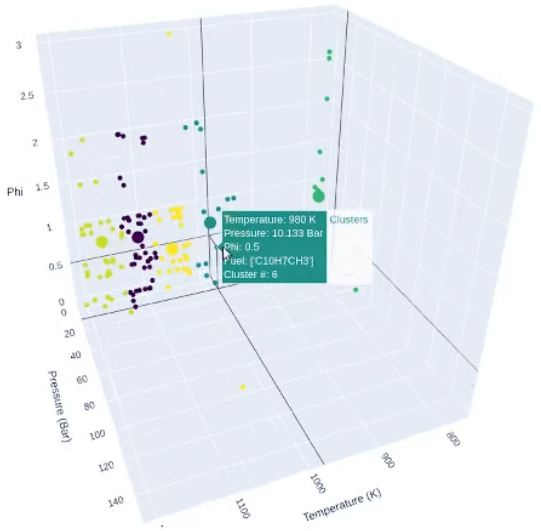

Figure 3: Here's a scatterplot of a model: each point represents an experiment, and the three axis (T, P, Phi) are the ones that I believed made the most sense to choose

K-means clustering The way K-means works is quite simple: it requires the number of desired clusters to be generated, and of course the variables to take into consideration when clustering. The "variables" are essentially the columns of the model. K-means calculates the euclidean distance between each point, then creates a cluster for each group of points with a relatively low distance between them.

2 Implementation

The program is written in Python and uses various data science libraries, such as Pandas, scikit-learn, Plotly and numpy.

The code can be found on my GitHub profile, at `https://github.com/pannuba/datamining-project-polimi`

2.1 Features

Correlation matrix The *getCorrelationMatrix* function takes the Pandas dataframe of the model and returns another dataframe that is the correlation matrix. It calculates the correlation coefficient of each pair of variables (columns), using the correct type of coefficient depending on whether the variables are numerical or categorical. If both variables are categorical it calculates Cramer's V, if one is numerical and one is categorical it finds the correlation ratio, and if both are numerical it calculates Kendall's Tau. Curiously, after using Python libraries to determine the distribution of each variable, I found out that no variable has a normal distribution, which is why I am not using Pearson's coefficient. Later in the code, it uses Plotly to plot a heatmap that is significantly easier to read than a raw table.

Permutations The *getPermutations* method of the Dataset class returns a list of dictionaries, where each dictionary is a permutation of (Experiment Type, Reactor, Target, Fuels). As explained earlier, the permutations are then used to filter the two datasets when comparing them. I chose to discard permutations with less than 10 total experiments, as the sample size would be too small to draw any meaningful conclusions.

Model comparison The program plots a bar chart where each bar represents the average score of a model's permutation, while also displaying the standard deviation. Each permutation has two adjacent bars, one for each model, which makes it easy to compare them. When calculating the average, median and standard deviation of the score for each permutation, I performed K-means clustering to make the calculation more accurate by discarding the experiments outside the biggest cluster.

Other There are other functions that act as a support to the other main functionalities of the program, such as a function that ranks the clusters by how many points they have, or one that simplifies the values of the temperature, pressure and phi by replacing the interval with an average, making calculations easier. There is also a function that allows to clearly visualize the clusters of a dataset in a 3D space, while also displaying the relevant information of each point.

3 Results

After running the program and passing it two models these are the results:

Correlation matrix When looking at the correlation matrix, it is clear that the main focus should be on the correlation between the score and the other variables, as one of the main goals of the model analysis is to find why and how the score increases or decreases in certain experiments. In figure 4, we can see that the score is correlated to the categorical columns (experiment type, reactor, target and fuels). This is not surprising, since we already know that a model is going to perform differently on different permutations of experiments.

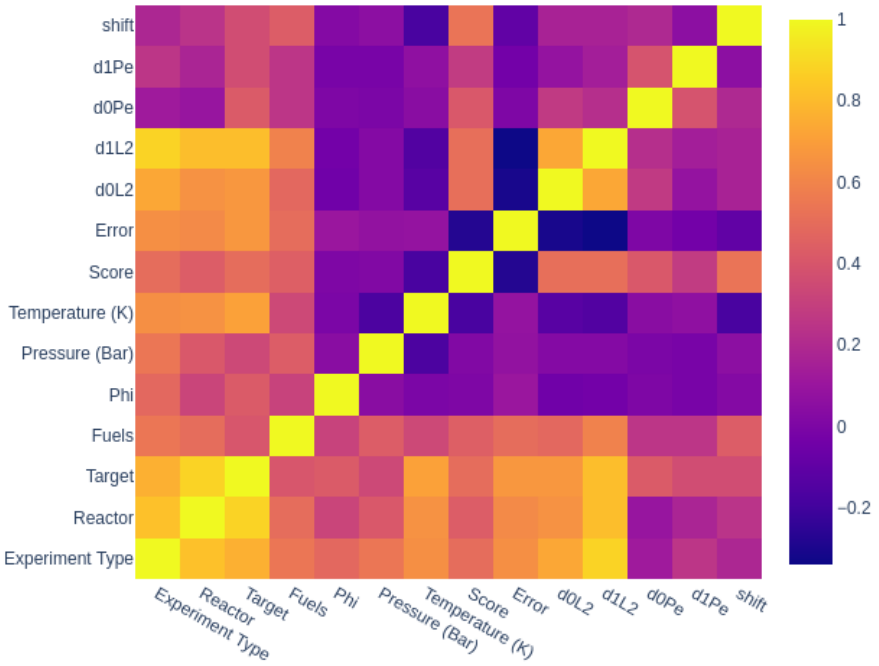

Figure 4: Heatmap of the correlation matrix of one model.

Bar chart Below is a figure of the bar chart that shows a clear comparison of the two models. Hovering each bar with the cursor shows the value of each categoric variable. It is very interesting that the four worst performing experiment types have two things in common: they all have "ignition delay measurement" as their experiment type, and "shock tube" as their reactor. Also, the target are all similar: it's tau_P(slope) in the worst one, and tau_OH(max or slope) in the other three.

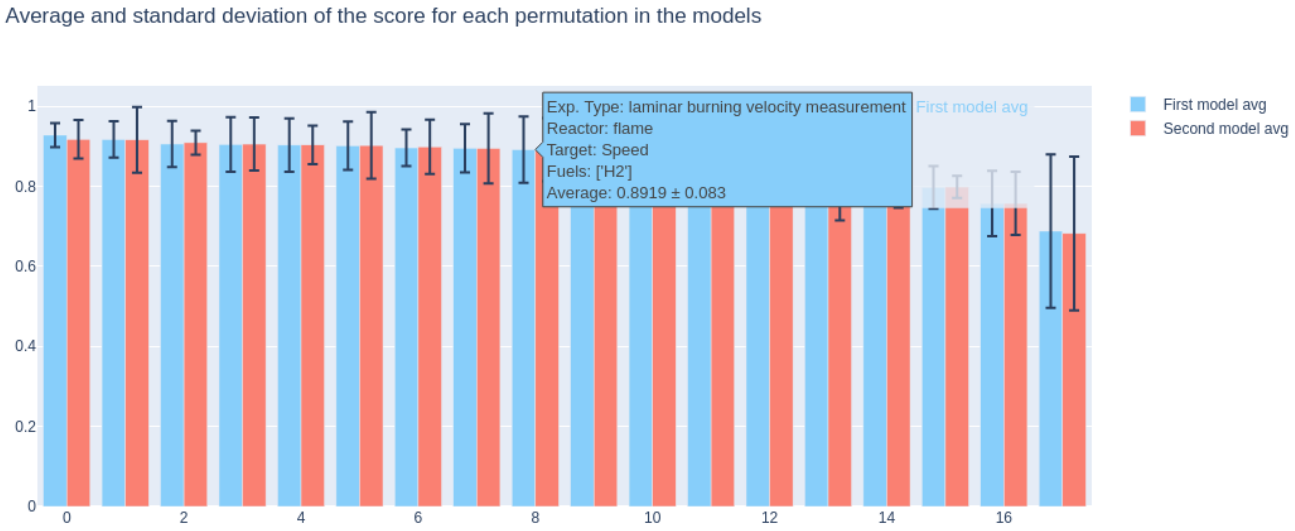

Figure 5: The bar chart comparing the models. The X axis is the permutation's index, the Y axis is the Score.

July 2022