STEM Games

Technology Arena

Day 2

14 May 2024, Umag, Croatia

Task 1:

Wastewater soft sensor

1.1 Introduction

Wastewater treatment is essential to protect public health. Untreated wastewater can contain harmful bacteria, viruses, and parasites that can cause serious illness, but also residual chemicals from different industrial plants. By removing these contaminants, wastewater treatment plants help to ensure that our water supplies are safe to drink and recreate in.

One wastewater treatment plant from Croatia implemented software sensors (soft sensors) and advanced process control for the purpose of increasing their treatment efficiency. Soft sensors are not physical instruments but computer models that estimate wastewater parameters based on existing data. This allows for continuous monitoring and optimization of treatment processes without the need for additional hardware sensors. Although soft sensors have a lot of potential in industrial application, the model on which it is based needs to be robust and accurate.

Process engineers of that wastewater treatment plant have concluded that this is not the case for their model. Namely, the displayed concentrations of sulfur compounds in the wastewater of chemical plants show much different values than those shown by the analysis carried out in the laboratory.

1.2 Problem Statement

Your team has been tasked with fixing this. Process engineers from the plant came up with the idea to solve the problem using artificial intelligence. They employed you to develop a new model from the data they have collected through two months of successful plant operation. They have not specified which machine learning technique you have to use, they left that decision to you.

Hint: there are two main branches of machine learning, supervised and unsupervised learning, you have been given data that allows you to take both paths, but be sure to correctly prepare your data.

1.3 Data

The wastewater treatment plant data is given in a CSV file and contains parameters described in the table below.

Parameter name	**Description**	**Data tip**
DT	Date and time	Boolean
TI1	Temperature of water inlet to rector 1, in Kelvin	Float
T1	Temperature in reactor 1, in Kelvin	Float
WF	Flow of water, in kg/h	Float
TI2	Temperature of water inlet to rector 2, in Kelvin	Float
T2	Temperature in reactor 2, in Kelvin	Float
TO	Temperature of water outlet, in Kelvin	Float
S	Concentration of sulfur compounds **predicted by the old soft sensor,** in ppm	Float

Example

Five example samples are shown below in table.

DT	**TI1**	**T1**	**WF**	**TI2**	**T2**	**TO**	**S**
01.09.2022 00:00	326.41	330.79	808.04	323.22	332.21	257.94	6.19
01.09.2022 00:01	326.45	330.78	806.73	323.21	332.32	257.95	6.2
01.09.2022 00:02	326.34	330.77	806.13	323.21	332.31	257.9	6.72
01.09.2022 00:03	326.23	330.76	806.57	323.19	332.19	257.84	7.05
01.09.2022 00:04	326.11	330.75	805.14	323.17	332.13	257.83	7.06

Data from the analytical laboratory is also given in a CSV file and contains parameters described in the table below.

Parameter name	**Description**	**Data tip**
DT	Date and time	Boolean
LS	Concentration of sulfur compounds **analytically determined,** in ppm	Float

Example

Five example samples are shown below in table.

DT	**LS**
1.9.2022. 4:30	12.9
1.9.2022. 14:00	11.7
1.9.2022. 23:56	12.7
2.9.2022. 4:30	11.2
2.9.2022. 14:00	9.3

1.3.1 Files

- plant_data.csv - contains wastewater plant data measurements
- LAB_data.csv - contains the actual sulfur concentration determined in an analytical laboratory
- Test_data.csv - this is the test data for which you need to submit your sulfur concentration predictions

1.4 Evaluation

Leaderboard

One of the most important aspects of Kaggle Competitions is the Leaderboard. The Competition leaderboard has two parts.

The public leaderboard provides publicly visible submission scores based on a representative sample of the test data. This leaderboard is visible throughout the competition.

The private leaderboard, by contrast, tracks model performance using the remainder of the test data. The private leaderboard thus has final say on whose models are best, and hence, who the winners and losers of the Competition will be. Which subset of data is calculated on the private leaderboard or a submission's performance on the private leaderboard is not released to users until the competition has been closed.

Many users watch the public leaderboard closely, as breakthroughs in the competition are announced by score gains in the leaderboard. These jumps in turn motivate other teams working on the competition in search of those advancements. **But it's important to keep the public leaderboard in perspective. It's very easy to overfit a model, creating something that performs very well on the public leaderboard, but very badly on the private one. This is called overfitting.**

In the event of an exact score tie, the tiebreaker is the team which submitted earlier. Kaggle always uses full precision when determining rankings, not just the truncated precision shown on the Leaderboard.

Solutions will be ranked by the obtained RMSE value. Max. 30 points.

Root Mean Squared Error (RMSE)

RMSE stands for Root Mean Squared Error. Like MSE, it measures the average squared difference between the predicted values and the actual values. However, RMSE takes the square root of the MSE, which makes it easier to interpret because it is in the same units as the dependent variable.

RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Task 2:

Data for solar energy production

2.1 Introduction

The decarbonization goals and energy self-sustainability of each country prompts an increase of renewable energy sources such as hydro energy, solar energy, and wind energy. Consequently, the increase of renewable energy sources requires finding of appropriate energy storage solutions, including novel techniques such as hydrogen storage. Solar panels are an appropriate solution for renewable energy harvesting since they can easily be installed even at private houses.

To find the appropriate location for the solar power plant installation it is required to find the location with the best conditions. Three brothers decided that they would invest in solar power plants at the roof of their houses, however they can invest in only one location.

Luckily, they were able to obtain the historical data of solar radiation for locations of their houses. This data can help them identify which location would be the most appropriate one.

2.2 Problem statement

The problem is that input data has some missing information. It is required to **determine best ML model that will predict Ghi value for next 3 months with most appropriate features** so the best location for installing solars can be determined.

Hints: Evaluate appropriate treatment of missing data (elimination of these records, replacing missing data with some substitute value (imputation methods), etc.)

Explore dimensionality reduction. Features can be omitted, manipulations with the data can be made (e.g. PCA), etc.

2.3 Data

The meteorological data is given in a CSV file for 3 separate locations and contains parameters described in the table below.

Name	Units	Description
air_temp	°C	**Air Temperature** The air temperature at 2 meters above surface level.
albedo	0 - 1	**Albedo Daily** The average daytime surface reflectivity of visible light, expressed as a fractional value between 0 and 1. 0 represents complete absorption. 1 represents complete reflection.
azimuth	°	**Solar Azimuth Angle** The angle between the horizontal direction of the sun, and due north, with negative angles eastwards and positive values westward. Varies from -180 to 180. A value of -90 means the sun is in the east, 0 means north, and 90 means west.
clearsky_dhi	W/m2	**Clearsky Diffuse Horizontal Irradiance (DHI)** The diffuse irradiance received on a horizontal surface (if there are no clouds). Also referred to as Diffuse Sky Radiation.
clearsky_dni	W/m2	**Clearsky Direct Normal Irradiance (DNI)** The diffuse irradiance received on a horizontal surface (if there are no clouds). Also referred to as Diffuse Sky Radiation.
clearsky_ghi	W/m2	**Clearsky Global Horizontal Irradiance (GHI)** Total irradiance on a horizontal surface (if there are no clouds). The sum of direct and diffuse irradiance components received on a horizontal surface, in the clear sky scenario, (i.e. no water or ice clouds in the sky).
clearsky_gti	W/m2	**Clearsky Global Tilted Irradiance (GTI)** Total irradiance received on a surface, if there are no clouds, with defined tilt and azimuth (sum of direct, diffuse and reflected components), fixed or tracking. Use Utility Scale Sites for the most accurate GTI (In the clear sky scenario, i.e. no water or ice clouds in the sky), because it is based on detailed PV system specifications including detailed tracker behavior.

cloud_opacity	%	**Cloud Opacity** The attenuation of incoming sunlight due to cloud. Varies from 0% (no cloud) to 100% (full attenuation of incoming sunlight).
dewpoint_temp	°C	**Dewpoint Temperature** The dewpoint temperature at 2 meters above surface level.
dhi	W/m2	**Diffuse Horizontal Irradiance (DHI)** The diffuse irradiance received on a horizontal surface. Also referred to as Diffuse Sky Radiation. The diffuse component is irradiance that is scattered by the atmosphere.
dni	W/m2	**Direct Normal Irradiance (DNI)** Irradiance received from the direction of the sun. Also referred to as beam radiation.
ghi	W/m2	**Global Horizontal Irradiance (GHI)** Total irradiance on a horizontal surface. The sum of direct and diffuse irradiance components received on a horizontal surface.
gti	W/m2	**Global Tilted Irradiance (GTI)** Total irradiance received on a surface with defined tilt and azimuth (sum of direct, diffuse and reflected components), fixed or tracking.
precipitable_water	kg/m2	**Precipitable Water** Precipitable water of the entire atmospheric column.
precipitation_rate	mm/h	**Precipitation Rate** Precipitation rate in millimeters per hour. An estimate of the average precipitation rate during the selected period, expressed in millimeters per hour - not an accumulated value.
relative_humidity	%	**Relative Humidity** The relative humidity at 2 meters above ground level. Relative humidity is the amount of water vapor as a percentage of the amount needed for saturation at the same temperature. A value of 50% means the air is 50% saturated.
surface_pressure	hPa	**Surface Pressure** The air pressure at surface level.

snow_depth	cm	**Snow Depth** a measure of the physical snow pack on the ground, measured in centimeters .
snow_soiling_rooftop	%	**Snow Soiling Loss - Rooftop** Loss in rooftop PV module (DC) production loss due to snow soiling. 0% means no snow soiling losses. 100% means snow is fully covering all modules.
snow_soiling_ground	%	**Snow Soiling Loss - Ground Mounted** Loss in ground mounted PV module (DC) production loss due to snow soiling. 0% means no snow soiling losses. 100% means snow is fully covering all modules.
snow_water_equivalent	cm	**Snow water equivalent** is the snow depth liquid equivalent.
wind_direction_100m	°	**Wind Direction 100m** Wind direction at 100m above ground level. Zero means true north. Varies from 0 to 360. A value of 270 means the wind is coming from the west
wind_direction_10m	°	**Wind Direction 10m** Wind direction at 10m above ground level. Zero means true north. Varies from 0 to 360. A value of 270 means the wind is coming from the west
wind_speed_100m	m/s	**Wind Speed 100m** Wind speed at 100m above ground level.
wind_speed_10m	m/s	**Wind Speed 10m** Wind speed at 10m above ground level.
zenith	°	**Solar Zenith Angle** The angle between the direction of the sun, and the zenith (directly overhead). The zenith angle is 90 degrees at sunrise and sunset, and 0 degrees when the sun is directly overhead.

The data comprises hour measurements for the period of 3 years. The data consist from all available meteorological data relevant for solar insolation for chosen locations. It is known that GHI component of solar irradiation is most important for determination of solar potential for solar panels, therefore it is required to predict Ghi value for the period of next 3 months for 3 chosen locations. Some examples of data are shown in table below.

air_temp	albedo	azimuth	clearsky_dhi	clearsky_dni	clearsky_ghi	cloud_opacity	dewpoint_temp	dhi	dni	ghi	precipitable_water
6	11	-15	0	0	0	821	49	0	0	0	163
6	1	-44	0	0	0	907	51	0	0	0	164
6	1	-65	0	0	0	933	54	0	0	0	167
7	1	-79	0	0	0	899	58	0	0	0	164
6	1	-91	0	0	0	877	56	0	0	0	16
6	1	-101	0	0	0	795	52	0	0	0	16
6	1	-111	0	0	0	714	51	0	0	0	162
6	1	-121	1	1	1	647	48	0	0	0	158
6	1	-132	36	231	65	566	48	29	0	29	15
6	1	-144	67	498	181	634	49	65	0	65	147
6	1	-157	81	627	277	779	47	61	0	61	154
6	1	-171	91	662	329	826	45	57	0	57	159
6	1	174	95	648	331	786	45	71	0	71	158
6	1	159	91	592	283	773	45	64	0	64	167
6	1	146	72	504	196	791	46	40	0	40	187
6	1	134	43	299	85	621	46	34	0	34	20

2.3.1 Files

- bakar.csv – meteorological data for location 1
- cavle.csv - meteorological data for location 2
- drenova.csv - meteorological data for location 3

2.4 Evaluation

Solutions will be ranked by the obtained RMSE value. **The data for the last 3 months in the dataset will serve for the purpose of this evaluation.** It is necessary to evaluate RMSE for each location. **The best solution will be the model that has the smallest RMSE for all 3 locations, cumulatively.** I.e. the best solution will be the cumulative RMSE (the sum of 3 RMSE values, one for each location). For each location the same machine learning algorithm should be used, but trained for every location separately. For example, if you want to use Linear regression, the linear regression with same model parameters should be done 3 times, one time for each location. For each location the RMSE should be calculated and summed up.

Root Mean Squared Error (RMSE)

RMSE stands for Root Mean Squared Error. Like MSE, it measures the average squared difference between the predicted values and the actual values. However, RMSE takes the square root of the MSE, which makes it easier to interpret because it is in the same units as the dependent variable.

RMSE is calculated as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Task 3:

Point-Assignment Equation Modeling

3.1. Introduction

A faculty at the University of Zagreb practices granting generous monthly stipends whose recipients are determined at the beginning of each academic year. While all students are automatically taken into consideration, the choice will fall on only 5% of the most suitable according to a final ranking. The ranking is performed by assigning points, which are expressed as real numbers in range [0, 1000]. Their values are computed by a simple algorithm, which is essentially a linear combination of scores achieved on various criteria.

A recent employee, Bob, was assigned the simple task of running the ranking program and informing the selected students before a government-imposed deadline. Figuring that clicking the run button and sending out a couple of e-mails would be easy-peasy, he procrastinated until the end of the working hours the last day before the deadline. Sipping the last gulp of now-cold afternoon coffee, he clicks on the link to the web-service hosting the algorithm. Alright – now, we only need to enter the password. The password!?

Bob had completely forgotten about that! And the only people who can reset the password for him are Information Support employees, who have already gone home!

- I can't believe it – Bob muttered while going frantically through the pages of a binder full of previous student records and their achieved points – do I really now have to reconstruct the whole grading algorithm before midnight?!

3.2. Problem Statement

Help Bob reconstruct the grading algorithm which will return the calculated points based on input student data. Because students do not need to enter any additional information, he is certain that all data needed to achieve perfect accuracy is already available in faculty databases. However, he needs to figure out the equation, including possible data transformations. Additionally, to keep the algorithm more manageable and easier to work with, Bob has decided to **keep the parameter count as low as possible**.

3.3. Data

Below is an example table containing a snippet of possible input data. All students have an unique`id`, along with a large number of other columns with different relationships with the point count. Keep in mind that a student is competing for funding every year of studying. All columns are available in dataset provided.

id	**year**	**first_name**	**last_name**	**course_grade_1**	**...**	**deans_list**
12345678	2020	Ana	Anić	5	...	Y
87654321	2023	Ivo	Ivić	4	...	N

3.3.1 Example Training Output

The training predictions contain three data columns: `id`, `year` and `points`, where the `id` isthe unique student identifier and `year` signifies the year that specific student achieved the assigned `points`.

id	**year**	**points**
12345678	2020	905.4
87654321	2023	610.8

3.3.2 Example Output

The output predictions should contain two data columns: `id` and `points`, where the `id` is theunique student identifier and `points` their predicted points as computed by the algorithm.

id	**points**
12345678	905.4
87654321	610.8

3.3.3 Files

- old_student_data.csv - contains all student data **except** their points.
- old_points.csv - contains points computed for each student each year.
- student_data.csv - this is the test data for which you need to submit your predictions.

3.4. Evaluation

Solutions will be ranked by the L0-regularized RMSE value.

$$\mathrm{RMSE}(\mathbf{w}) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \mathbf{w}^T\mathbf{x}_i)^2}$$

$$\mathcal{L}(\mathbf{w}) = \mathrm{RMSE}(\mathbf{w}) + \lambda\|\mathbf{w}\|_0$$

Mean Squared Error (MSE) is a metric representing the average squared difference between the estimations and the true values. With the difference being squared, errors in prediction of, for example, 5 and -5 are penalized equally, as well as e. g. -0.5 and 0.5. A higher MSE is interpreted as bigger error, signifying a less precise computation algorithm.

Root Mean Squared Error (RMSE) is a metric computed by taking the root value of the MSE.

Regularization methods are widely used in ML to keep the model simpler, which often means having less parameters or their coefficients being lower. There are several ways to implement regularization, from stopping the training process earlier to penalizing less suitable models through loss function.

In this case, we employ L0-regularization of the (RMSE) loss function, where the model is, alongside the error represented through RMSE, punished by its total parameter count (i. e. **L0- norm**) multiplied by a constant λ. Here, you can assume **λ = 1**.