Predicting Water Quality Index (WQI): methodologies and results

PREZIOSA ALESSIA (590012)

Politecnico di Bari - xTech bip

Project overview

The project aims to use Machine Learning techniques to predict a water quality index based on data collected from India.

Water quality has implications for public health, ecosystem sustainability, and socio-economic development. Since India is world's second most polluted country, it's paramount to have a water quality prediction to be ahead of environmental problems and adopt preemptive strategies.

Two objectives will be pursued:

1. Understanding **water quality dynamics**: during Exploratory Data Analysis, relationships between variables (environmental factors and water quality parameters) will be detected to have a grasp of the aquatic ecosystems. **Is there a detectable trend? Is there a correlation between variables?**
2. Predictive **modelling**: during Modelling, predictive models will be developed to estimate water quality parameters based on input features to contribute to the advancement of water quality monitoring practices and to foster sustainable management of water resources in India.

The produced tools will be used to facilitate early detection of potential contamination and provide support for policymakers and environmental agencies.

Data understanding

The dataset, *obtained from a government agency responsible for environmental monitoring and regulation* and collected from monitoring stations across 18 states of India, encompasses water quality measurements.

It's a 48 KB dataset of 534 rows and 11 columns:

- **STATION CODE**: Unique *identifier* for each monitoring station;
- **LOCATIONS**: *Name* or description of the monitoring station location;
- **STATE**: The *state* in which the monitoring station is located;
- **TEMP**: *Temperature* of the water (°C);
- **DO**: *Dissolved Oxygen* levels in the water (mg/L);
- **pH**: *pH* level of the water;
- **CONDUCTIVITY**: *Electrical conductivity* of the water (μS/cm).
- **BOD**: *Biochemical Oxygen Demand* of the water (mg/L).
- **NITRATE_N_NITRITE_N**: *Combined concentration of nitrate and nitrite* in the water (mg/L).
- **FECAL_COLIFORM**: *Concentration of fecal coliform bacteria* in the water (CFU/mL).
- **TOTAL_COLIFORM**: *Concentration of total coliform bacteria* in the water (CFU/mL).

To compute the **label** for **water quality**, a method involving a **Water Quality Index (WQI)** calculation is employed. It is determined by summing the product of the aggregated quality ratings (Qn) and the unit weights (Wn) assigned to the following parameters: DO (0.281), pH (0.165), CONDUCTIVITY (0.009), BOD (0.234), NITRATE_N_NITRITE_N (0.028), FECAL_COLIFORM (0.281).

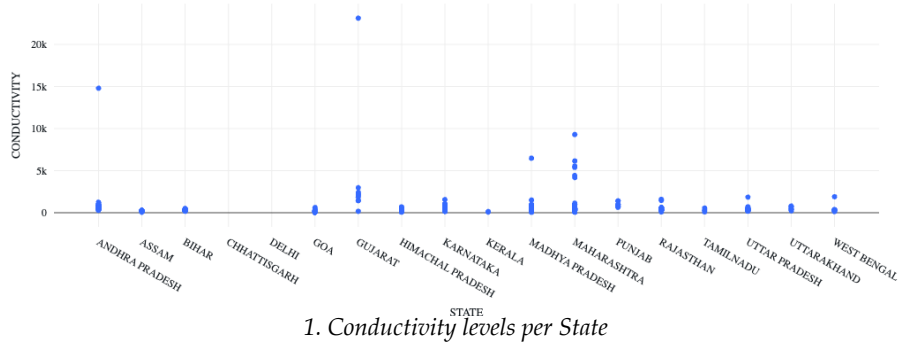
The csv dataset is read with spark.read.format using a manual schema defined as in the following:

STATION CODE: integer
LOCATIONS: string
STATE: string
TEMP: double
DO: double
pH: double
CONDUCTIVITY: double
BOD: double
NITRATE_N_NITRITE_N: double
FECAL_COLIFORM: double
TOTAL_COLIFORM: double

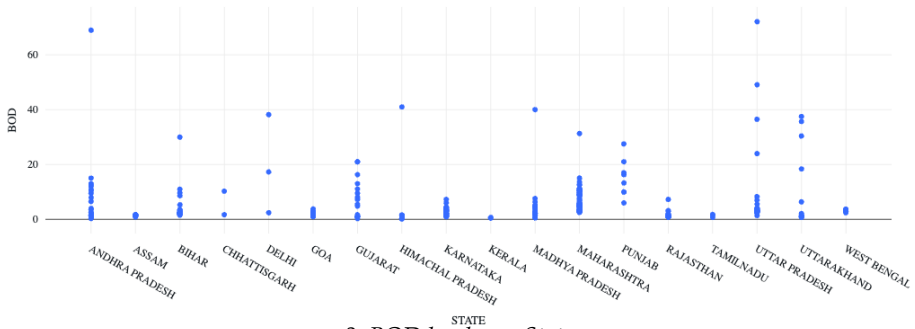
Data preparation

Before computing the label, we need to be sure that no attribute is missing and there are no mistakes in collected data.
Dataset is randomly split into 2 subsets: train (70%) and test (30%). The train set is used for analysis while the test one will be used for performance prediction.

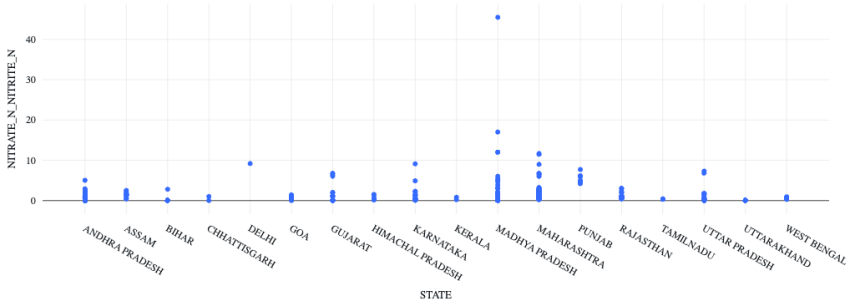
From a first statistical analysis on train dataset, it is shown that values of skewness are high for Conductivity (9.81), Concentration of nitrates (8.65), BOD (4.58) and Coliforms (6.99): this may be because there are some outliers (as visible in the following graphs) but, **except for the pH column where a value of 14.7 is recorded (pH ∈ [0, 14]), we assume other variables don't present mistakes so no other record will be deleted.**



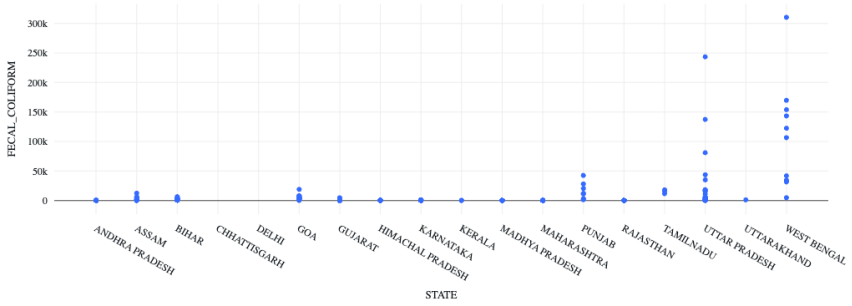
1. Conductivity levels per State



2. BOD levels per State



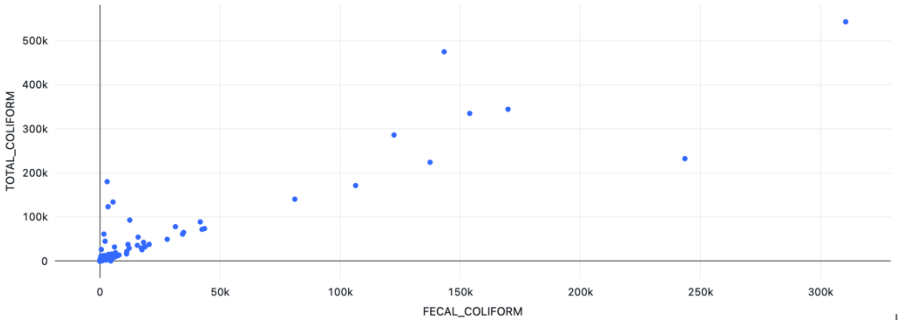
3. Concentration of nitrates per State



4. Fecal Coliform values per State

Data cleaning and transformation

It results that all the null (missing: 32) values of TOTAL_COLIFORM have a corresponding null (missing) in FECAL_COLIFORM; they are also strictly correlated with a Pearson coefficient of 0.915 (so, all information provided by TOTAL_COLIFORM is already provided by FECAL_COLIFORM). Since FECAL_COLIFORM is used to compute the label, TOTAL_COLIFORM won't be used to predict the WQI.



5. Total Coliform and Fecal Coliform

Missing values

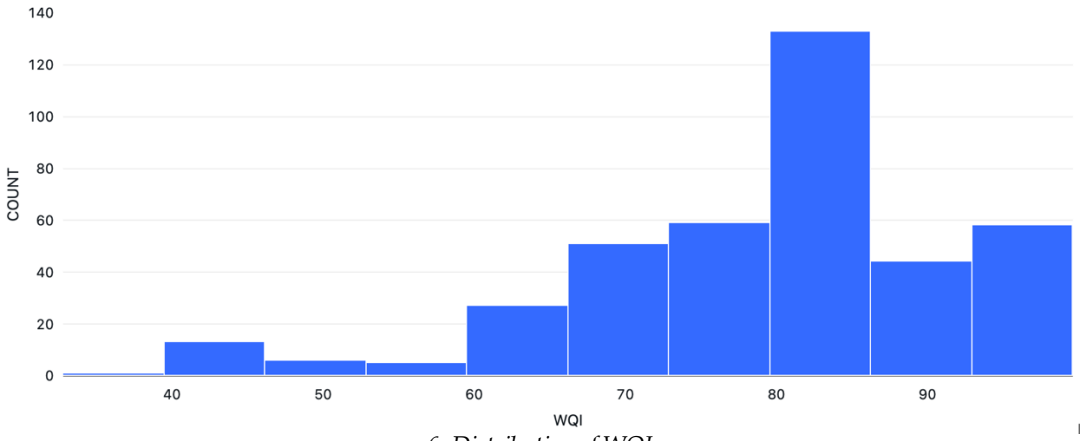
As regards the most interesting (numerical) columns (**'TEMP', 'DO', 'pH', 'CONDUCTIVITY', 'BOD', 'NITRATE_N_NITRITE_N', 'FECAL_COLIFORM'**), we assume values are missing completely at random (MCAR: refers to a situation where the probability of a data point being missing is completely unrelated to both observed and unobserved data). The variables with the highest percentage of missing data are Fecal Coliform (16.58%) and Conductivity (5.03%); others assess around 1% or lower.

Since the dataset is quite small, deletion of records is not taken into consideration:

- An imputing method (mean substitution) will be fitted on the (numeric) training variables and applied on the testing one: we can run analysis as if all cases are complete, but it reduces variability.
- We'll include an additional field (a dummy column) that acts as an indicator for missing value (1 = value is missing, 0 = value is observed). It uses all information about missing values, but it results in biased estimation. The missing indicators will be included in the algorithm.

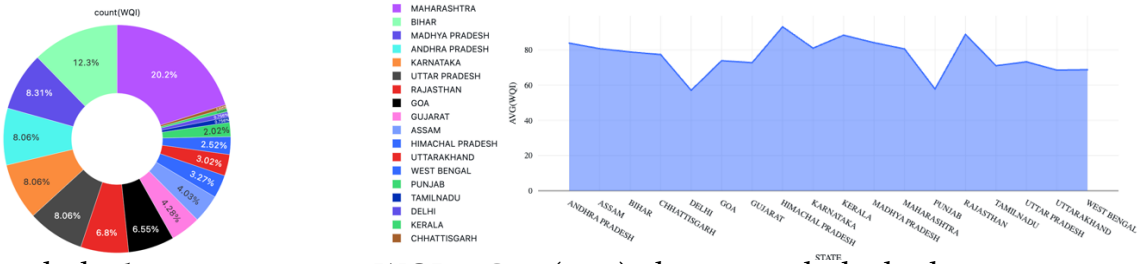
In order to do so, Imputer (it's an Estimator) is used: it implements the method fit() which accepts a DataFrame and produces a Model.
The Model (it's a Transformer) implements the method transform() which accepts a DataFrame and produces another DataFrame appending one or more columns.
In this case the transformation is as simple as it gets: missing values for each predictor variable are substituted with the mean (computed as if nulls were not present) of the variable itself.

Once dataset is clean, we can **compute the labels** for each observation.



6. Distribution of WQI

A simple Statistical Analysis of the WQI is conducted; it is visible that the dataset is not completely balanced: most of the observations are contained in the range [79.6, 86.3], as a matter of fact the distribution of Water Quality Index is negatively skewed (-1.0).

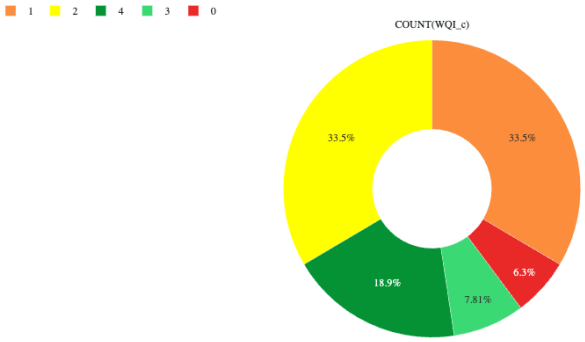


The State with the lowest minimum WQI is **Goa** (32.8), the one with the highest maximum WQI is **Himachal Pradesh** (99.62), which is also the state with the highest average WQI (93.2). The most polluted state is **Delhi** (57.1, even though there are only 3 observations for it)

The State with the highest number of observations is **Maharashtra** (80), the one with the lowest are **Chhattisgarh** and **Kerala** (2).

The WQI is discretized to create classes[1] of Use:

0. [0, 59]: Poor quality
1.]59, 79]: Bad quality
2.]79, 84]: Medium quality
3.]85, 89]: Good quality
4.]90, 100]: Excellent quality



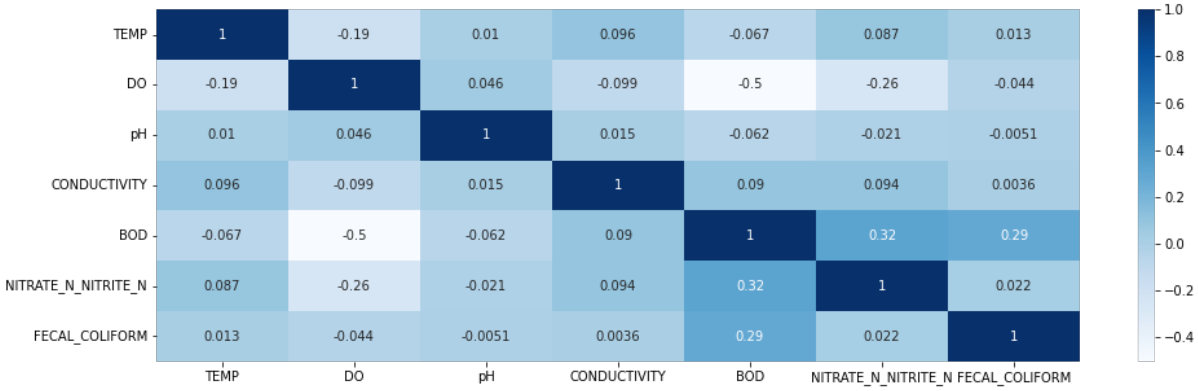
7. Distribution of WQI discretized in 5 classes

Most of the observations belong to classes 1 and 2 (Bad-Medium quality)

Exploratory Data Analysis: understanding water quality dynamics

We're going to explore the relationships between different environmental factors and water quality parameters to gain insights into the complex dynamics of aquatic ecosystems.

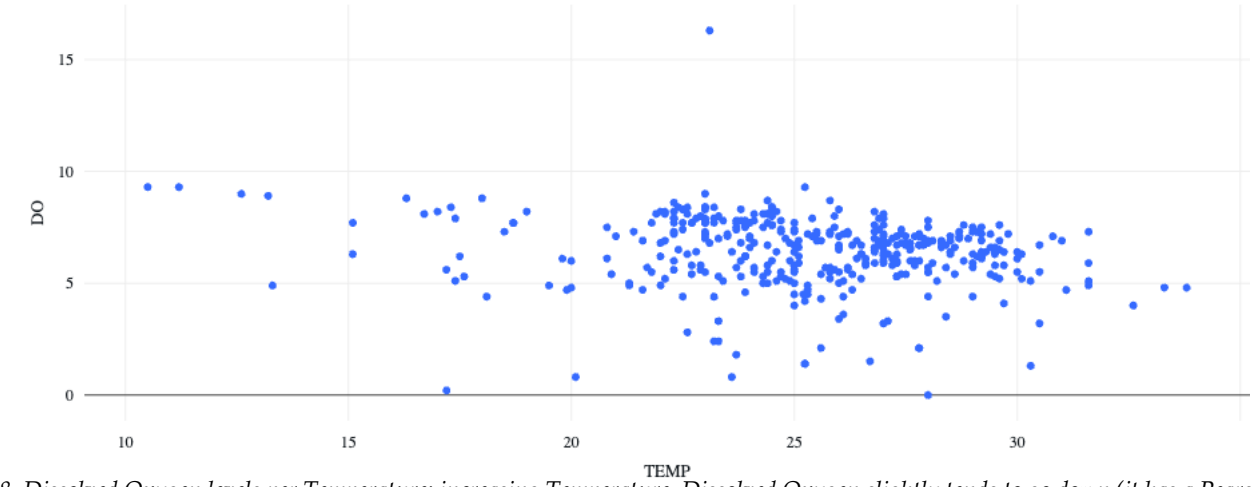
Correlation is computed between predictor variables:



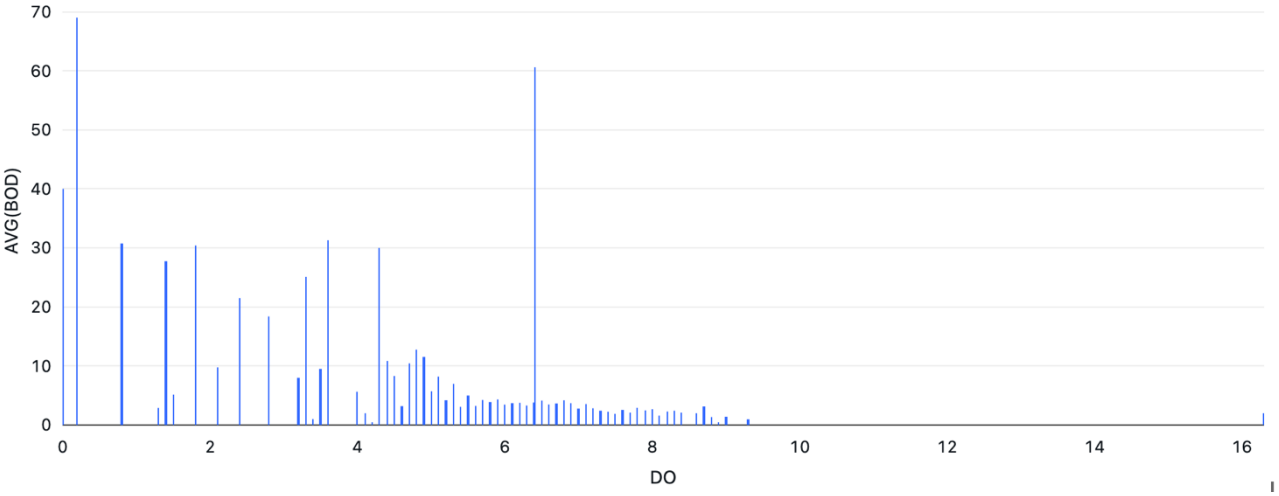
Even though the Pearson coefficient is indicative exclusively of a linear relationship, it is a measure of correlation: all relationships whose variables have a correlation coefficient > 0.1 or a correlation coefficient < -0.1 are investigated.

[1] Division in classes is not backed up by scientific papers

Below, the most visible ones are shown:

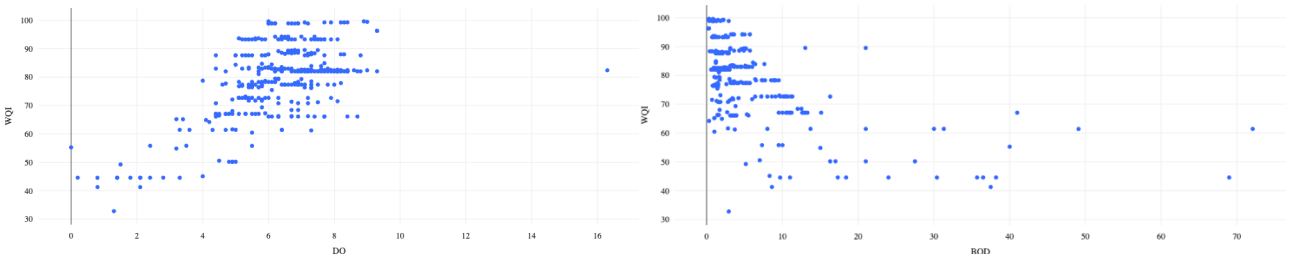


8. Dissolved Oxygen levels per Temperature: increasing Temperature, Dissolved Oxygen slightly tends to go down (it has a Pearson coefficient of -0.19)



9. Average values of BOD per DO: there's a slight inverse relationship with a Pearson coefficient of -0.5

Since the one between BOD and DO is the strongest relationship, we try to gain more insights out of it using WQI.



It's visible that as BOD increases, WQI decreases (Pearson: -0.56); vice versa with DO (as DO increases, WQI increases; Pearson: 0.61). As a matter of fact, DO and BOD are inversely proportional.

Modelling

A bit of feature engineering is done before fitting models on data.

To handle different variables, a VectorAssembler is used: it merges the numeric features into one.
Next step is Normalisation, very different ranges of values might badly influence predictions and algorithms' performance: to avoid this kind of problem, a StandardScaler is used. The StandardScaler standardizes features by removing the mean and scaling to unit variance (Z-score normalization): $x_i = \frac{x_i - \mu_x}{\sigma_x}$. Dummy variables (missing indicators) aren't scaled.
Final step is merging dummy variables and scaled ones into a final single feature.

Algorithms

The proposed problem can be solved with supervised learning algorithms.

We could adopt different algorithms, for example, **Linear Regression** (it will be used to predict a continuous variable: WQI) and/or **Random Forest** (it will be used to predict a discrete variable: WQI_c)

Linear Regression is a supervised Machine Learning algorithm used for predicting a continuous target with one or more predictor variables. The algorithm learns the relationship between the features and the target while minimizing the difference (in terms of metrics like RMSE or R^2) between predicted and actual values.

Random Forests are ensembles of decision trees (each of them is trained on different samples of data). Randomness is also found in feature selection and data sampling. They are extremely robust (also against unscaled data) and reduce the risk of overfitting: it's the diversity of each tree to help improve the overall model performance and generalization on unseen data. They are used for binary and multiclass classification and for regression; moreover, they have no problem handling with categorical features (they don't need to be encoded).

So, a LinearRegression and a RandomForestClassifier models are built.

To guarantee hyperparameters tuning, **ParamGridBuilder** is used: it builds a parameters' grid for Grid-Search based model selection.

For the **regressor,** 2 parameters can be identified:

- regParam: it's the regularization parameter to prevent overfitting ([0.001, 0.01, 0.1, 0.2, 0.5])
- elasticNetParam: it's the ElasticNet mixing parameter, in range [0, 1]. For alpha = 0, the penalty is an L2 penalty. For alpha = 1, it is an L1 penalty ([0.0, 0.5, 1.0])

The **forest**, instead, is tuned on:

- maxDepth: maximum depth of each tree. ([10, 15, 20])
- numTrees: number of decision trees. ([15, 20, 25, 30])

The **Evaluators** are used, of course, to evaluate the performance of models. The best regressor will be chosen based on the highest R^2 while the best forest is chosen based on the highest F1.

Hyperparameters tuning is conducted via **Cross Validation** (5 folds), with the help of the previously defined grid of parameters.

The object is built as in the following:

```
pipeline = Pipeline(stages=[assembler, scaler, final_assembler, algorithm])

crossValidator = CrossValidator(
    estimator = pipeline,
    estimatorParamMaps = paramGrid,
    evaluator = evaluator,
    numFolds = 5,
    seed = 42
)
```

Algorithms are both fitted on train datasets.

Evaluation and conclusions

The testing set is eventually used to estimate the model on unseen data.
The regressor doesn't achieve a great performance; the following metrics are reached:

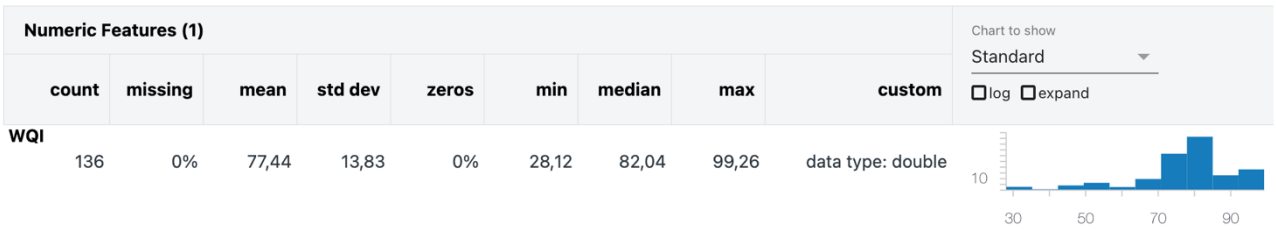
R^2	RMSE	MSE	MAE
0.55	9.21	84.80	6.63

The forest, on the other hand, reaches an almost optimal performance:

F1	Accuracy	Weighted Precision	Weighted Recall	Weighted TPR	Weighted FPR
0.8966	0.8970	0.8998	0.8971	0.8971	0.0413

It's clear the Random Forest achieves better results. It may be due to the fact that the target WQI_c has been discretized in 5 different classes while the regressor needs to obtain more precise values in order to achieve higher performances.

As regards the regressor, WQI's distribution of the test set and prediction's distribution are similar in terms of parameters but as visible in the histograms, the second one is more skewed.

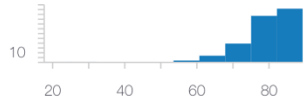


Numeric Features (1)

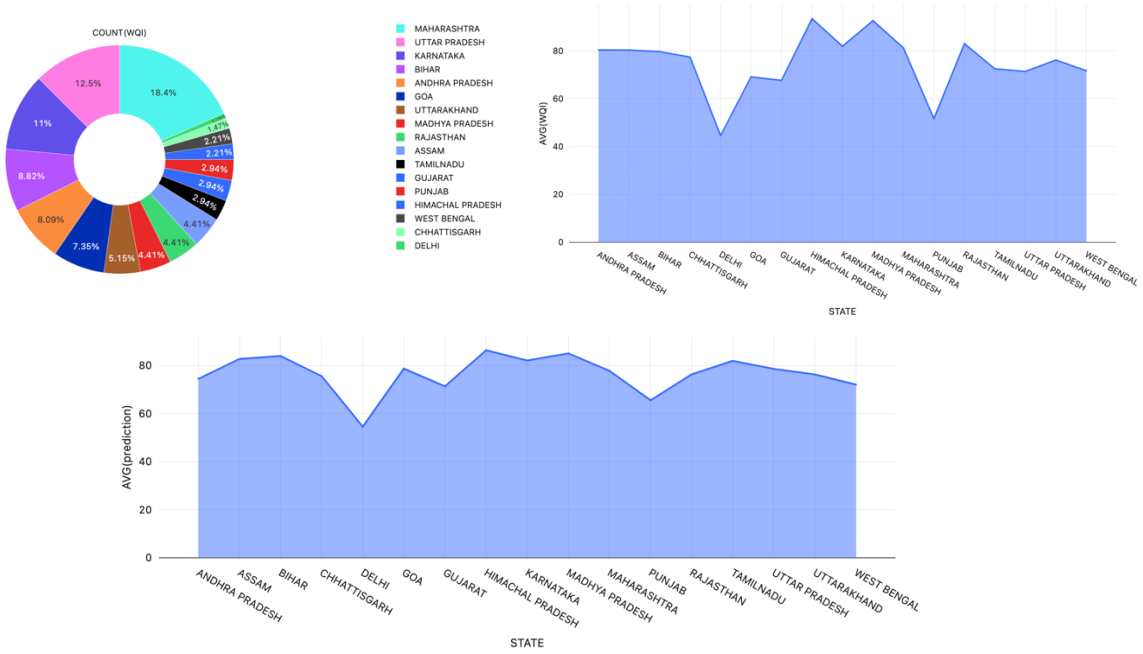
Chart to show: Standard

log expand

	count	missing	mean	std dev	zeros	min	median	max	custom
prediction	136	0%	78,54	9,55	0%	17,7	80,45	89,37	data type: double

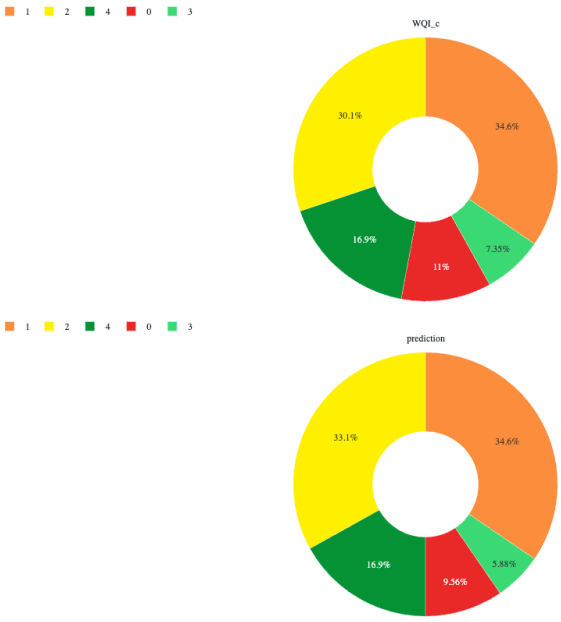


As the training dataset, also the testing one is not completely balanced: most of the observations are contained in the range [77.9, 85.0] but most of the predictions end up in the range [82.2, 89.4], as a matter of fact the predictions distribution of Water Quality Index is more negatively skewed (-1.33 < -3.17).



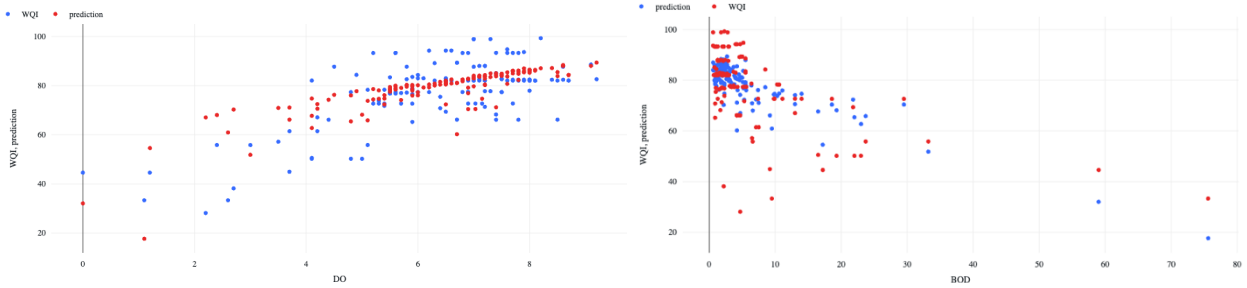
The State with the lowest minimum WQI is again **Goa** (28.1) for the real samples but it's **Uttar Pradesh** (17.7) for the predictions; the one with the highest maximum WQI is **Madhya Pradesh** for both real samples and predictions (99.26 and 87.99 respectively); **Himachal Pradesh** is the state with the highest average WQI for both real samples and predictions (93.4 and 86.3 respectively). The most polluted state is **Delhi** (only 1 observation is in the testing dataset)

The State with the highest number of observations is **Maharashtra** (25), the one with the lowest is **Delhi** (1). Note also that no samples from **Kerala** are in the testing dataset.



Also in the testing dataset, most of the observations belong to class 1 and 2 (see left) Percentages are quite similar (between real samples and predictions).

As regards distributions per State percentages are similar with respect to actual samples (graphs are not shown).
The states of Andhra Pradesh(11), Assam(6), Bihar(12), Goa(10), Gujarat(4), Karnatka(15), Madhya Pradesh(6) and Maharashtra(25) present slight differences in percentages



In the testing dataset, the inverse relationship is present. It's also possible to observe that predictions follow the correct trend.

Pearson coefficients in the following:

WQI/DO	Prediction/DO	WQI/BOD	Prediction/BOD
0.70	0.83	-0.58	-0.87

Coefficients are higher for predictions: the model learns the pattern.

In conclusion, what stands out the most from the analysis is that: Temperature doesn't affect the WQI (it was predictable since it hasn't been used to compute it); Dissolved Oxygen levels are higher for higher WQIs; pH is centered around 7; Conductivity, Biochemical Oxygen Demand, Concentration of Nitrates and Fecal Coliform tend to decrease as WQI gets higher.