**Environmental Sustainability Machine Learning Water Potability Project**

**By: Elizaveta Lyadova | Jessica Zimmerman | Mohamed Kassim | Beruchya Dao-Bai**

**Slide 1:**

[Title slide]

**Slide 2**

**J**: Environmental sustainability - pressing, we sit it in the crazy weather patterns, rising sea levels, polluted water

We felt it was important to understand water potability, perhaps enable more access to clean water

**Slide 3**

**B:** In line with our theme on environmental sustainability and water resources, we designed our model to show whether water is potable, or safe to drink based on a number of variables which we will share in the following slides. All in all our basic question can be summarized as: Is water potable based on a key number of chemical and physical properties of water, yes or no?

**Slide 4**

**B:** We found a dataset with 3276 different water samples. And this dataset was then set up in a Microsoft cloud database where a left join was performed on the index value of two tables.

The final database had the following features: pH, water hardness, solids present in the water, chloramines, sulfate, conductivity, carbon, trihalomethanes, turbidity, and a potability column. With that, we were able to carry out our model and ultimately add a potability prediction column based on our model

**Slide 5**

**M:** preprocessing of data included filling null values and ensuring that the right data types would be used

**Slide 6**

**M:** The better understand our data we chose to view them as histograms. For example, the pH histogram shows that most samples were around the normal water pH of 7

**Slide 7**

**J:** Identify ways to improve model accuracy

One select only features relevant to the model

Transform variables

Standard scalar to normalize variables with mean = 0, std dev = 1

Minimize the influence of large data values that might not be an accurate reflection of the entire dataset

**Slide 8**

**J:** Look at the importance of different variables

Solids and hardness are most important

**Slide 9**

**J:** Default train size is 75%, increased to 80% -> aiming to improve model accuracy

Stratified on “y” meaning we would preserve the proportion of our target in the train set

So it equates to the proportion present in the original dataset

**Slide 10**

**J**: The Balanced Random Forest Classifier creates smaller simpler decision trees trained on a random subset of features, a bunch of weak learners

These smaller trees are combined to form a strong learner which is usually the average prediction of all the weak learners

The prediction, in our case is, potable or not potable.

**Slide 11**

**J**: Benefits of this kind of model

Robust at overfitting because the smaller decision trees are each trained on a different subset of data

Can tune the algorithm easily with the n\_estimators input

The higher the input, the more decision trees the model creates, usually yields better performance

Limits

Feature importance may not tell the full story, maybe none of the features are that great at predicting potability

Model is easy to understand conceptually, but you really can’t control what the model does, the coefficients it creates

Creating more decision trees, adding massive datasets can strain your computer’s memory and also take a long time to render

**Slide 12**

**M[orange box]:** We attempted an experiment based on the thought that our accuracy could potentially be higher with a larger dataset. We artificially doubled the dataset by replicating each record twice to have a dataset of 6552 records. When we doubled the values no accuracy change happened.

**B [grey box]:** As expected we faced some challenges with setting up our model as Mohamed shared and having coherent values in the newly created prediction column. Our main issue is that we couldn’t figure out why our model only predicted all 0s (i.e. not safe to drink) or all 1s potable/safe to drink whether it was based on the original or artificially doubled dataset. We knew that we had to fix that and we started off by changing parameters in the algorithm like the n\_estimators values, train\_size, or the random\_state of selection. Nothing positive availed in the prediction column until we went back to removing the data scaling method we had used. I realized our dataset was not suited for scaling and that affected all the predictions by making them either exclusively potable or exclusively not potable. After realizing scaling was not necessary for our dataset, accuracy rose and predictions were varied.

NB: n\_estimators = number of trees in the forest to improve performance of model but it makes the model slower

Random\_state = control the random factor of selection in the training and testing aspect, basically making the train size not different everytime providing us with more stability to troubleshoot other areas

**Slide 13**

**M:** While we were troubleshooting we tested the fact that an increase in our dataset size could improve our accuracy, and so, we doubled our dataset and we tried two things. First thing is that we doubled the dataset and rerun the model the accuracy was still low at around 60%. The second attempt involved descaling the doubled dataset and that led to an accuracy of 92% which confirmed that scaling our dataset was an issue.

**Slide 14**

**B:** With a final score of 65%, we knew it was not perfect but we also realized where our model could improve based on the experiment we carried out with the provisional results by increasing the dataset size and descaling the dataset.This score right here is based on a descaled dataset and the original data source of 3000+ records. We will speak more about these lessons as we discuss the takeaways after viewing the dashboard on Tableau.

**Slide 15**

**J:** [Introduction] Here’s our dashboard and we used tools such as python, tableau and google slides to prepare the layout. We have included a tableau interactive map, pie charts, histograms, barcharts and a histogram

**J: Dashboard Graph 1**

Histogram of pH filtered by potable and nonpotable data, pH is displayed on the top

Non Potable - blue, potable - orange

Many potable and nonpotable samples with a pH of 7, which we know is the pH of water

So pH cannot be the only factor in determining water potability or we may need to resample our dataset

**Dashboard Graph 2**

0 not potable, 1 is potable - basic count of samples

Tells us we have enough potable samples to create predictions and test accuracy

**Dashboard Graph 3**

Average feature value filtered by potable and non-potable

Not much variation in average value based on potability

Large range in feature values - some single digits, others in thousands

May need to transform some variables

**M: Dashboard Graph 4 “pH predictions for potability”**

Similar to prior graph but displaying predictions instead of original potability, created in python and slightly cleaner

**M: Dashboard Graph 5**

Heatmap is displayed to highlight:

Heatmaps help to visualize data through variations in colouring. Heatmaps are good for showing variance across multiple variables, revealing any patterns, displaying whether any variables are similar to each other, and for detecting if any correlations exist in-between them (source: https://datavizcatalogue.com/methods/heatmap.html#:~:text=Heatmaps%20are%20good%20for%20showing,correlations%20exist%20in%2Dbetween%20them.&text=A%20legend%20is%20required%20alongside,it%20to%20be%20successfully%20read.)

**B: Dashboard Graph 6 pie charts**

With an accuracy score of 65% the predictions classified more values as potable than the original potability with a number of inaccurate predictions based on the accuracy score

**Slide 16**

**B:** Overall our modeling experience to determine whether water was potable, ‘yes’ or ‘no’ was interesting and informative. Our main takeaways are that our model stands a chance at being more accurate provided more time and resources. Some other lessons we learnt are the importance of selecting the right features to explore, for example in the case of water potability the chemical and physical properties of water such as pH, water hardness etc helped us determine water potability. Additionally, treating missing values like we did by filling null values with the mean of the column, and increasing the train\_size and n\_estimators to tune the algorithm helped make it more precise and accurate by 1-3% based on the different combinations. We also learnt that scaling a dataset may or may not work for every kind of dataset and that was really important in our case. And ultimately, we really think that our provisional results hinted at the fact that a larger dataset would significantly improve our chances of having more accurate scores.

Individual Team member challenges:

L: Tableau

J: Tableau & Machine learning

M: Machine Learning & Data processing

B: Database & Machine Learning

**Slide 17**

**B:** With that, we have come to the end of our presentation on modeling to predict water potability and we used tools such as GitHub, Python, Pandas, SQL, and Tableau to make our project happen.To tackle our challenge we worked collaboratively on most parts and also specialized for eg. Dataset cleaning and database work was Mohamed and I, Machine learning: Mohamed and Jessica, and I supported with troubleshooting and tuning the algorithm, Dashboard: Jessica and Liza on Tableau, and myself on the design, python graphs and layout, and finally on the presentation aspect we had Liza & Jessica on content creation and organization, and I helped with the overall design and content as well. Thank you so much for you attention. We hope you have enjoyed our presentation.

**Slide 18**

**[last slide]**