Data Cleaning and Preprocessing

Data Science II (COSC 4337)

Dr. Ricardo Vilalta

Submitted by:

Brayan Gutierrez (1865588)

Jericka Ledezma (1968730)

Katie To (2026435)

1. *Data Description*

The dataset was found on the Open Data Telangana portal under the title of "Telangana Ground Water Department - Post monsoon Water Quality Data". The Open Data Telangana portal is a website hosted by the Telangana government in order to support their 'Open Data Policy', as such the portal is the central repository of all the datasets of the Government of Telangana that should be in the public domain. A link to the dataset can be found here: https://data.telangana.gov.in/dataset/telangana-ground-water-department-post-monsoon-water-quality-data .

The data was collected and compiled by the The Telangana Ground Water Department of the Telangana government in the state of Telangana, India from various districts and villages in the area. Water samples were pulled from 34 districts in total. Furthermore, within these districts, water samples were pulled from cities within mandals as well.

The dataset was composed of five files, one for each year from 2018 to 2022. For this project, we are only interested in the most recent data, as such we selected only the file from 2022 to be used in this project.

Column Name	Data Type	Description
sno	int64	An abbreviation for "sample number". It is identical to the row number and/or record number of the row.
district	object	A significant administrative division in India.
mandal	object	An intermediate administrative division that is smaller than a district but larger than a village. It consists of a group of villages.
village	object	A village is the smallest administrative unit and represents a small community or a cluster of settlements.
lat_gis	float64	The latitude of the location where the sample was pulled from.
long_gis	float64	The longitude of the location where the sample was pulled from.
RL_GIS	float64	The ground water level of the water sample.

season	object	The year and whether or not it is post or pre-monsoon season.
pH	float64	A measure of how acidic or basic the water is on a scale of 0 (acidic) to 14 (basic).
E.C	float64	An abbreviation for “Electrical Conductivity”. A measurement of water’s ability to conduct electricity. Measured in microsiemens per centimeters.
TDS	float64	An abbreviation for “Total Dissolved Solids”. These can be organic and inorganic substances such as minerals, salts, metals, cations, or anions dissolved in water. Measured in parts per million (PPM) and milligrams per liter (mg/L).
CO3	float64	A measurement of the amount of Carbonate (CO3) in the water. Measured in parts per million (PPM).
HCO3	float64	A measurement of the amount of Bicarbonate (HCO3) in the water. Measured in parts per million (PPM).
Cl	float64	A measurement of Chloride in the water. Measured in mg/L.
F	float64	The amount of fluoride in the water. Measured in mg/L.
NO3	float64	The amount of nitrate (measured as nitrogen) in the water. Measured in mg/L.
SO4	float64	The amount of sulfate in the water. Measured in mg/L.
Na	float64	The amount of sodium in the water. Measured in mg/L.
K	float64	The amount of potassium in the water. Measured in mg/L.
Ca	float64	The amount of calcium in the water. Measured in mg/L.
Mg	float64	The amount of magnesium in the water. Measured in mg/L.
T.H	float64	An abbreviation for “Total Hardness”. A measurement of the hardness of the water,

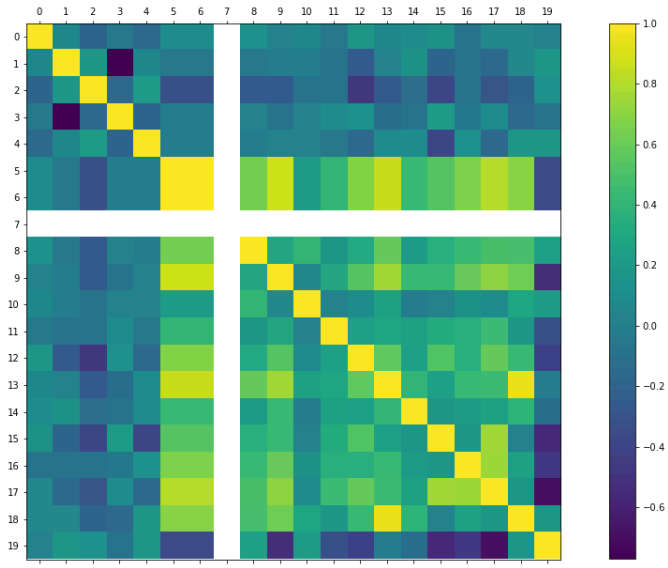
		otherwise the concentration of limestone and other dissolved minerals in the water. Measured in mg/L.
SAR	float64	An abbreviation for “Sodium Absorption Ratio”. It relates the amount of sodium relative to calcium and magnesium in water. Expressed by: SAR = [Na]/(([Ca]+[Mg])/2)1/2.
Classification	object	The water’s quality class. Classified based upon two qualities: their salinity level (C). and sodium content (S) on a scale of 1 (low) to 4 (very high).
RSC meq / L	float64	The Residual Sodium Carbonate index of the water. Measures the alkalinity hazard if used in irrigation.
Classification.1	object	The target class. A final classification dictating whether the water is safe (P.S.), marginal (MR), or unsafe (U.S.).

2. Exploratory Data Analysis

While the districts the water was collected from seems to be evenly dispersed with an average 25 from each district, the Nalgonda district had 79 samples collected from it. At the very least, it seems each district had at least 10 samples taken from it, with the amount collected ranging from 11-79 samples. Approximately 750 samples have a final classification of P.S., making safe water the majority. 37 of the samples are U.S., and 30 of the samples are MR. However, the water quality class majority is at C3S1, which indicates that 480 samples have a high salinity level with a low sodium level. The next most occurring water quality class is C2S1 with 253 samples, which is not too far behind from the leading class. The World Health Organization (WHO) recommends a TDS level of less than 300 mg/L for drinking water with the absolute limit for drinking water being 600-900 mg/L, and the average TDS in the dataset is 683.96 mg/L. While the average is within the standards, the range staggers greatly from 77-1534 TDS mg/L. From this, we can say that there is a trend in which the majority of water being classified safe is most likely water with an average salinity level and a low sodium content (https://www.netsolwater.com/ideal-tds-level-in-water.php?blog=896).

The outliers are mostly seen in any of the minerals contained in the water rather than the location or classifications as seen above. An image of all the boxplots and histograms can be found in the attached Jupyter Notebook under the “Outlier Detection” section. The shape of these histograms are skewed right, meaning that outliers are sudden spikes or increases in the water contents. The only ones that are not skewed right are pH and RSC that are more bell-shaped. These outliers in the water content can be caused

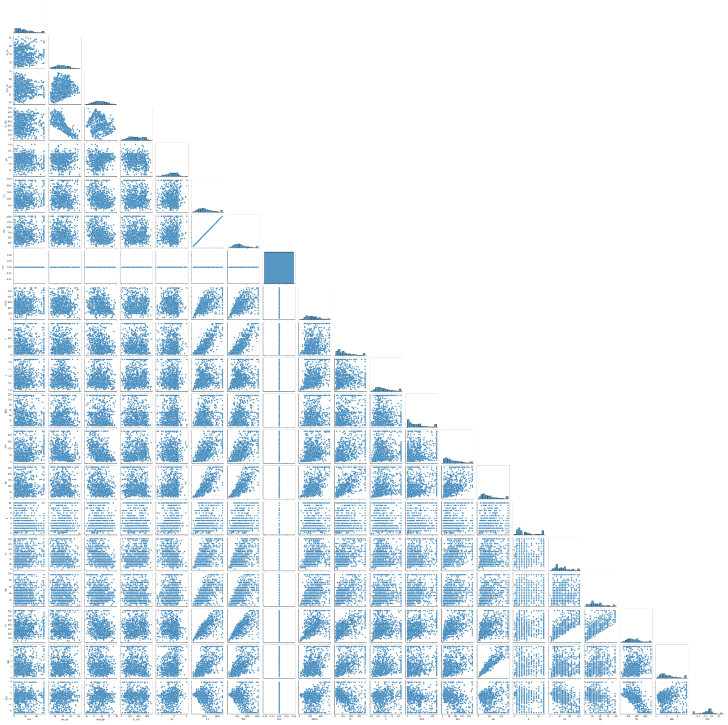
by outside factors that we are not privy to, such as illegal dumping, sewer overflows, or other ways of water pollution that can drastically change the water to cause these outliers. These have been treated with flooring, as described in the data cleaning process below, and should not have a heavy impact on the outcome and classifications.



The numeric correlation matrix helped us get a preliminary understanding before moving into the feature engineering on the relationships that are exhibited between the features. Features five and six in the numeric dataset are the only features that have a relationship symbolized with the yellow being the highest mark for correlation. Feature five represents 'Electrical Conductivity (EC)' and feature six represents the "Total Dissolved Solids (TDS)" in a water sample. This is a significant observation because the electrical conductivity is influenced by the presence of dissolved salts and minerals in the water, and since this measurement is quantified in feature six it's understandable for there to be a strong relationship between both the features. The relationship between EC and TDS also presents similarities with features 8-18 since they both register the same color with these features.

The pair plot is a unique way to analyze correlation with all the features, but also gives a better understanding of how a feature performs when it is compared to itself. In this plot CO_3 displays how it's a feature to consider eliminating in the feature engineering phase because it does not correlate with any

feature and is displayed as a horizontal/vertical line in all the plots it is compared with. Potassium (K) exhibits an interesting relationship with the other features that causes all the data points to not overlap and form columns on the graph. This could be due to the fact there is no linear overlap, and many of the datapoints are too close to truly depict a relationship on the plot. Furthermore, linear relationships are presented through the pair plot. E.C and TDS demonstrate positive linear relationships with TH, SAR, O, HCO_3, and Na. Based on this similarity SAR and Na share a strong positive linear relationship. Lastly, RSC and TH share a negative linear relationship.



3. Feature Engineering

In order to properly clean the data, we first sought to find any missing data. While reading the data we decided to drop any NA rows that were present in the data which reduced the number of rows from 1025 to 817. After reading the data, using the .info() method from the pandas library in order to print all information about the dataframe, we determined that there was no data missing from this dataset as all rows returned the same amount for each column.

After confirming that there was no more missing data, our next step was to address any outliers, if any were found. In order to detect outliers, a box plot and a histogram was constructed of each numeric

feature to find any abnormalities. To address the outliers, the dataset was floored at the 25th percentile and capped at the 75th.

Next, we removed the data that was irrelevant. A code was written to weed out the columns whose rows were 99% the same value, as the value of those columns would not be useful for the final classification if the features had the same value for every row. We used this to determine that the "season" and "CO3" features were unnecessary, thus we opted to remove them entirely.

With the data cleaned, we began our feature selection process. So that we could have multiple observations to validate our final selection of features, each team member opted to try a different supervised method, with the target class being "Classification.1" across all three methods. Before any method began, however, the entire dataset was standardized.

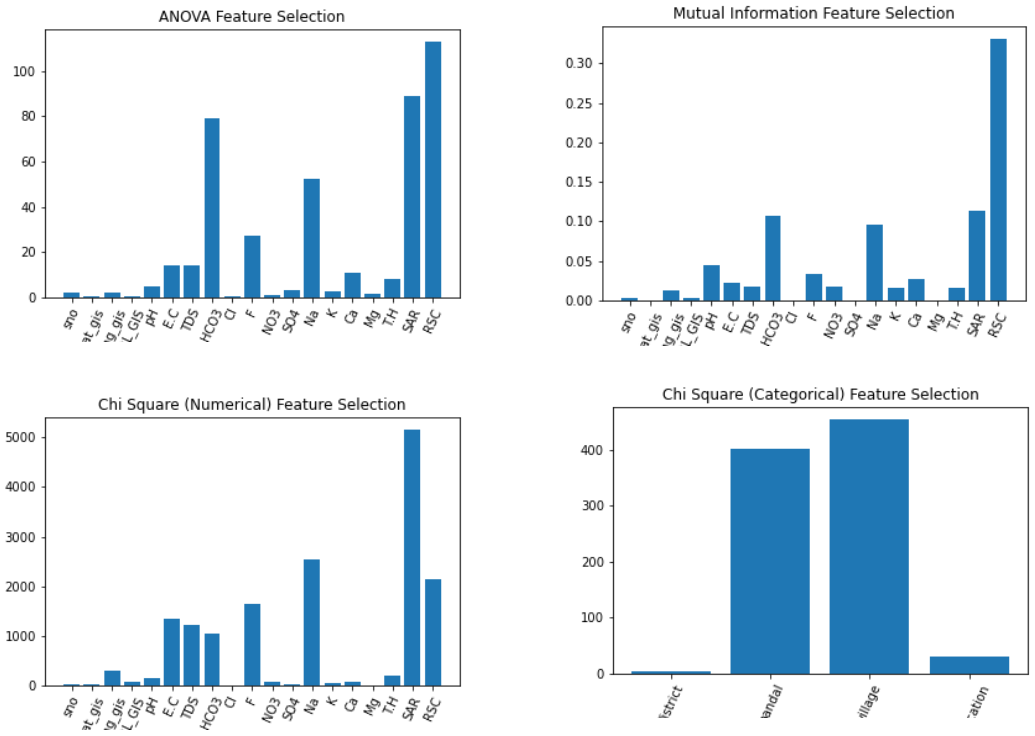
The first method was conducted by Brayan, who used the ANOVA method (https://machinelearningmastery.com/feature-selection-with-numerical-input-data/). This method calculates the F-statistic and p-value of each feature with the target variable, in this case "Classification.1" is our target variable. In this method, features that have a high F-statistic and low p-value are considered more relevant, while low F-statistic and high p-value are less relevant. In our notebook, we split the dataset with 60% training and 40% testing. We then fit the ANOVA function with the training data that consisted of all the numerical features and the target variable. Then we used that fitted ANOVA function on the testing data to find which features are most relevant, and plotted their scores to easily visualize the relevant features. We found that the relevant features were RSC, SAR, Na, E.C, TDS, HCO3, pH, Mandal, and Village.

The second method was the Mutual Information (https://machinelearningmastery.com/feature-selection-with-numerical-input-data/), which was also performed by Brayan. This method is useful when dealing with high-dimensional datasets. As our dataset has 26 columns and 1024 observations, it is high-dimensional so this method is very useful. This method also measures the dependency between variables based on the amount of information gained about the target variable, "Classification.1," based on the value of a particular feature. In this method of feature selection, features with high mutual information scores are considered more relevant, and are more useful in predicting the target variable. For our run of the Mutual Information method, similar to ANOVA, we used the numerical variables and found that the relevant features were RSC, SAR, Na, E.C, TDS, HCO3, pH, Mandal, and Village.

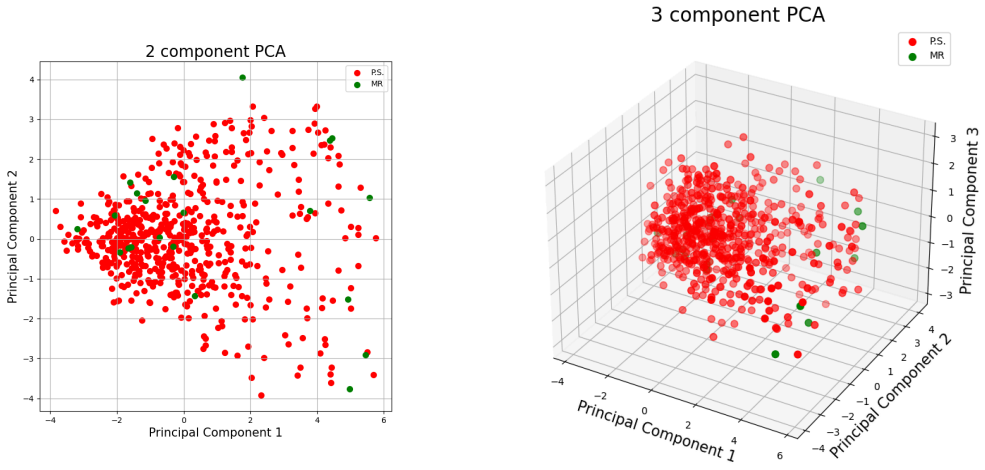
The third method was conducted by Jericka, who used the Recursive Feature Elimination method. After the initial cleaning and eliminating of empty rows it became clear that the amount of features would

increase the complexity of the models. The data science blog (https://www.analyticsvidhya.com/blog/2023/05/recursive-feature-elimination), Analytics Vidhya, mentioned using Recursive Feature Elimination as effective technique to systematically select the most important features from the dataset based on the specified amount to select. In our implementation of Recursive Feature Elimination, we utilized the scikit-learn library in Python to streamline the feature selection process. By creating a base model using logistic regression and specifying the desired number of features to select, we initialized the RFE object and fit it to the dataset containing features (X) and the target variable (y). This allowed us to rank the features based on their importance and identify the top features that optimize the model performance.

The fourth and final method was conducted by Katie, who used the Chi Square method. For this method, an outside website (https://machinelearningmastery.com/feature-selection-with-categorical-data/) was referenced to better understand how the process worked, and their code is modified for use in our dataset. The Chi Square method was run three times: once using only numerical features, once with only non-numerical features, and once with all. Each time, the data was prepared by splitting it into two test and train sets for the targets and features, which were then generated into encoded test and train sets with the use of ordinal encoding for the features to encode variables into integers and label encoding on the target set. In the end, we had two encoded training and testing sets for the features and the target for a total of four, which were then used with scikit-learn's SelectKBest defined with the chi2() function to transform the train and test sets and observe the dependencies between the target and the features. This will return the most desirable features.



Each method was made into a bar chart in the end, to better visualize across all three methods the results of each feature selection. After careful examination, the following features were kept: RSC, SAR, Na, E.C, TDS, HCO3, pH, Mandal, and Village.



Using the chosen numerical features, we performed principal component analysis of our already standardized data, and tested the first two principal components. We noticed that the unsafe (U.S.) class

was completely dropped and only the safe (P.S.) and marginal (MR) classes were kept. Although the results shown in the “2 component PCA” seemed to perform well, it only captured 70.86% of the variance. Even though that met the bottom threshold of how much variance was captured, we believed we should add one more principal component to capture more variance. We then ran another PCA with 3 components instead which can be seen in the “3 component PCA” 3D plot. This gave us better results with it capturing 83.66% of the variance while also reducing the dimensionality. With those results, we decided that 3 principal components is sufficient for this project.

4. Jupyter Notebook Instructions

The notebook we used is organized with labeled sections indicating the purpose of the code. Each section is delineated by a title followed by a line underneath. It is essential to have the CSV file of the dataset in the same directory as the notebook for successful execution. Failure to have both files in the same directory will result in the notebook failing to run. After the initial run of the notebook, any attempt to execute code before the "Unnecessary Data" subsection will result in an error due to the removal of columns in that section. To re-run sections preceding the "Unnecessary Data" subsection, the code needs to be run from the beginning or by selecting Kernel -> Restart & Run All. However, any code executed after this problematic section can be run multiple times without encountering errors. Also, after running the PCA plots, do not run any other plots again as that will cause issues. Make sure you have all the modules imported, including the ‘ipympl’ module.