|  |  |
| --- | --- |
| **Technical Report** | **Michaela Lambert** |
| **Water Quality Analysis** | |
| **URL to dataset:** https://www.kaggle.com/datasets/mssmartypants/water-quality?resource=download | |

|  |
| --- |
| **Overview** |
| **Problem Domain**: |
| Water quality is a critical environmental and public health issue. Poor water quality can lead to serious health problems, including waterborne diseases, and negatively impact ecosystems. The dataset on water quality provides various parameters such as pH, hardness, solids, chloramines, sulfate, and organic carbon levels, which are essential indicators of water safety and cleanliness.  According to the World Health Organization (WHO), approximately 2.2 billion people globally do not have safely managed drinking water services. Contaminated water is estimated to cause 485,000 diarrheal deaths each year. Monitoring these parameters helps in early detection of contaminants, ensuring safe drinking water, and preventing health issues.  Analyzing this dataset using statistical and machine learning methods can help in predicting water quality, identifying contamination patterns, and making data-driven decisions for water treatment and management. These methods can uncover insights that traditional monitoring might miss, providing a robust approach to managing water quality efficiently.  Reference:  WHO. (2019). Drinking-water. World Health Organization. Retrieved from  <https://www.who.int/news-room/fact-sheets/detail/drinking-water> |
| **Objective**: |
| The primary objective of this analysis is to assess the quality of water based on various physical and chemical parameters provided in the dataset. We aim to identify patterns and anomalies in the water quality data to ensure it meets safety standards.  **Questions:**   1. **What are the key indicators of poor water quality in this dataset?**    * *Approach:* I will use statistical analysis and visualization to identify the main parameters affecting water quality. 2. **Can we predict water quality based on the given parameters?**    * *Approach:* We will employ the Naive Bayes classification model to build a predictive model for water quality classification. 3. **How do the different parameters correlate with each other?**    * *Approach:* Using correlation analysis, we will explore the relationships between different water quality metrics.   **Methods:**   * **Naive Bayes Model:** This algorithm will be utilized to predict water quality categories based on the dataset's features. The Naive Bayes model is chosen for its simplicity and efficiency, making it suitable for this classification task. * **Correlation Analysis:** This will help in understanding the interdependencies between various water quality parameters, providing deeper insights into the factors that influence water quality.   By leveraging the Naive Bayes model and correlation analysis, we aim to provide a reliable mechanism for assessing water quality and ensuring it meets safety standards. |
| **Analysis** |
| **Exploratory Analysis**: |
| The dataset I sourced from Kaggle provides comprehensive measurements related to water quality, encompassing a variety of chemical and biological variables. These variables include concentrations of aluminum, ammonia, arsenic, barium, cadmium, chloramine, chromium, copper, fluoride, bacteria, lead, nitrates, nitrites, mercury, perchlorate, radium, selenium, silver, and uranium. Each of these variables is quantified, although the units of measurement are not specified within the dataset. Additionally, the dataset features a binary indicator, is\_safe, which denotes whether the water is considered safe (1) or not safe (0) based on the concentrations of the measured contaminants. This dataset likely results from sampling and laboratory analysis of water from various sources, aiming to assess and ensure water safety. The is\_safe variable serves as a crucial summary measure, providing a clear indication of the overall water safety status based on the detected levels of different contaminants.  The World Health Organization (WHO) sets guidelines for drinking water quality, identifying key contaminants that are most common and pose significant health risks worldwide. The following variables from my dataset are most commonly monitored and highlighted by the WHO: arsenic, lead, cadmium, mercury, nitrates and nitrites, bacteria and viruses, uranium, and chloramine. I will use these top contaminants identified by WHO as my key variables.  **Exploratory Data Analysis (EDA)**  **Correlation Analysis**  To understand the relationships between the variables, I conducted a correlation analysis and visualized the results using a heatmap. The heatmap showed the pairwise correlations between all variables, including the target variable is\_safe. This analysis revealed the following key insights:   1. **Cadmium**: Strong negative correlation with is\_safe (-0.26), indicating higher levels are associated with unsafe water. 2. **Arsenic**: Moderate negative correlation with is\_safe (-0.12). 3. **Viruses**: Moderate negative correlation with is\_safe (-0.10). 4. **Uranium**: Slight negative correlation with is\_safe (-0.08). 5. **Nitrates**: Slight negative correlation with is\_safe (-0.07).   **Statistical Comparison**  I performed t-tests to compare the mean levels of contaminants between safe and unsafe water samples. The results highlighted significant differences for the key contaminants:   1. **Cadmium**: Mean level in safe water is 0.017081, compared to 0.046115 in unsafe water. The t-statistic is -23.681126, with a p-value of 7.043962e-120, indicating a highly significant difference. 2. **Arsenic**: Mean level in safe water is 0.074624, compared to 0.172659 in unsafe water. The t-statistic is -11.114589, with a p-value of 1.721815e-28. 3. **Viruses**: Mean level in safe water is 0.226451, compared to 0.341871 in unsafe water. The t-statistic is -8.717387, with a p-value of 3.422658e-18. 4. **Uranium**: Mean level in safe water is 0.039002, compared to 0.045402 in unsafe water. The t-statistic is -6.780435, with a p-value of 1.283335e-11. 5. **Nitrates**: Mean level in safe water is 8.705680, compared to 9.962612 in unsafe water. The t-statistic is -6.463264, with a p-value of 1.084746e-10.         **A screenshot of a graph  Description automatically generated**  **Conclusion**  From the exploratory data analysis and statistical comparison, it is evident that cadmium, arsenic, viruses, uranium, and nitrates are key indicators of poor water quality. These contaminants are significantly higher in unsafe water samples. Conversely, contaminants such as aluminum and chloramine are more prevalent in safe water samples, suggesting they are associated with better water quality.  These findings align with the guidelines provided by the WHO, reinforcing the importance of monitoring these key contaminants to ensure water safety. By focusing on these high-risk contaminants, we can better understand the factors that contribute to unsafe water and take appropriate measures to mitigate these risks. |
| **Preprocessing**: |
| The preprocessing steps undertaken for the water quality dataset were crucial to ensure the accuracy and reliability of subsequent analyses. Initially, the dataset was loaded from a CSV file, containing various physical and chemical parameters essential for assessing water quality. Recognizing the importance of data integrity, the columns ammonia and is\_safe were converted to numeric data types. This step was necessary to handle any non-numeric values that could have been present due to data entry errors, ensuring that these critical columns were in the correct format for analysis.  Handling missing values was the next vital step. Any rows containing missing values were removed, which is essential to maintain the dataset's completeness and prevent bias that missing values might introduce. With a clean dataset, attention was turned to the outliers. Outliers in the aluminum and arsenic columns were identified using the Interquartile Range (IQR) method and replaced with the mean values of their respective columns. This approach mitigates the potential skewing effect outliers can have on statistical analyses and models.  Outliers in the nitrites column were handled differently; instead of replacing them, they were removed using the IQR method. This step was crucial to ensure that extreme values did not distort the analysis. The dataset was then narrowed down to relevant features based on new findings. Selected columns included indicators of poor water quality (cadmium, arsenic, viruses, uranium, nitrates) and better water quality (aluminum, chloramine), focusing the analysis on the most critical parameters.  To ensure consistency in data analysis, the selected features were standardized to have zero mean and unit variance. Standardization is vital in machine learning as it ensures that all features contribute equally to the model, preventing any single feature from disproportionately influencing the results. Finally, the standardized features were combined with the target variable is\_safe, creating a robust dataset ready for further analysis. This comprehensive preprocessing approach laid a solid foundation for accurately assessing water quality and deriving meaningful insights. |
| **Model Fitting**: |
| To fit the Naive Bayes model for predicting water quality, we begin by loading and cleaning the dataset. This involves reading the data from a CSV file and converting relevant columns to numeric types, handling any non-numeric entries by coercing them to NaN, and then dropping rows with missing values. Next, we handle outliers in key features by replacing them with the mean value if they fall outside 1.5 times the interquartile range. This ensures that extreme values do not skew the model's predictions.  Following data cleaning, we select key variables relevant to water safety, such as levels of cadmium, arsenic, viruses, and other contaminants. We then split the cleaned dataset into training and testing sets to evaluate the model's performance on unseen data. The features are standardized to have a mean of zero and a standard deviation of one, ensuring that all features contribute equally to the model.  With the data prepared, we initialize the Gaussian Naive Bayes model and train it using the training data. The model learns the probability distributions of each class (safe or not safe) given the features. After training, we use the model to make predictions on the test set and evaluate its performance using accuracy and a classification report, which provides metrics such as precision, recall, and F1-score. This process ensures a thorough understanding of the model's predictive capabilities and the key factors influencing water quality. |
| **Results** |
| **Model Properties:** |
| After fitting the Naive Bayes model, it's essential to understand its components and characteristics to fully leverage its predictive power. The Gaussian Naive Bayes model assumes that features follow a normal distribution and calculates the probability of each class based on the conditional probabilities of the features. The primary components of the fitted model include prior probabilities, means, variances, and likelihoods.  The prior probabilities represent the initial probability of each class (safe or not safe) before considering any evidence. These are derived from the distribution of classes in the training data. The mean values for each feature and class indicate the central tendency of feature values within each class, crucial for characterizing the Gaussian distribution. Variance values for each feature and class measure the spread of feature values around the mean, indicating variability. Higher variances suggest more variability in feature values for that class.  The model uses these means and variances to compute the likelihood of observing the feature values given each class. This likelihood, expressed as the joint log likelihood, helps the model determine the posterior probabilities during prediction, which are then used to classify new samples. By understanding these components, we can interpret the model’s decisions and assess its reliability. Summarizing model properties, such as prior probabilities, means, and variances, provides insights into the distribution and variability of features within each class, aiding in refining the model and improving its performance in predicting water quality. |
| **Output Interpretation**: |
| The final output of the Naive Bayes model for predicting water quality shows an overall accuracy of 91.37%, which indicates that the model correctly classified 91.37% of the test samples as either safe or not safe. The classification report provides a detailed breakdown of the model's performance across the two classes. For the 'not safe' class (0), the model achieved a precision of 0.95, a recall of 0.95, and an F1-score of 0.95, based on 1396 samples. This high precision and recall imply that the model is highly effective at identifying water samples that are not safe, minimizing false positives and false negatives in this class.  For the 'safe' class (1), the model achieved a lower precision of 0.67, a recall of 0.64, and an F1-score of 0.65, based on 203 samples. This indicates that while the model can identify some safe water samples, it is less reliable in this regard compared to identifying unsafe samples. The lower recall for the 'safe' class means that the model misses a significant number of safe water samples, which could lead to unnecessary caution regarding water safety.  In relation to the stated objective of assessing water quality and ensuring it meets safety standards, the model successfully identifies unsafe water samples with high accuracy and precision. However, the performance could be improved for identifying safe water samples to achieve a more balanced assessment. Overall, the model provides a reliable mechanism for monitoring and managing water quality, with room for further optimization to enhance its predictive capabilities for the 'safe' class.  A screenshot of a computer  Description automatically generated |
| **Evaluation**: |
| The Receiver Operating Characteristic (ROC) curve illustrates the performance of the Naive Bayes model in distinguishing between 'safe' and 'not safe' water samples. The curve plots the True Positive Rate (sensitivity) against the False Positive Rate (1-specificity) at various threshold settings. The presented ROC curve shows a high level of performance, with an area under the curve (AUC) of 0.83, indicating the model's good ability to differentiate between the two classes. Specifically, an AUC of 0.83 suggests that the model has a strong predictive capability, significantly better than random guessing, which would have an AUC of 0.5.  The True Positive Rate (TPR) on the y-axis indicates the proportion of actual 'safe' water samples correctly identified by the model, while the False Positive Rate (FPR) on the x-axis shows the proportion of 'not safe' samples incorrectly classified as 'safe'. The curve's shape, bowing towards the upper left corner, signifies a strong performance, as the model quickly captures the majority of true positives with a low FPR.  In conclusion, the ROC curve and AUC score of 0.83 confirm that the Naive Bayes model effectively predicts water quality. Despite some limitations in precision and recall for the 'safe' class, the model's overall performance is robust, meeting the objective of accurately assessing water quality based on various parameters. This capability is crucial for ensuring water safety standards are met and guides further improvements in the model and data handling processes.  A graph of a line  Description automatically generated with medium confidence |
| **Conclusion** |
| **Summary**: |
| The primary objective of this analysis was to assess the quality of water based on various physical and chemical parameters to ensure it meets safety standards. The Naive Bayes model was employed to predict water quality by classifying samples as either 'safe' or 'not safe'. The key findings indicate that the model effectively differentiates between these two classes, demonstrating strong predictive capabilities.  Through the analysis, it was found that the model accurately identifies 'not safe' water samples, which is crucial for public health and safety. The ROC curve further confirmed the model's robustness with an AUC score of 0.83, indicating a high level of performance in distinguishing between 'safe' and 'not safe' water samples.  These findings underscore the model's utility in monitoring and managing water quality, providing a reliable mechanism to support regulatory compliance and protect public health. The insights gained from this analysis can guide future improvements in water quality assessment, ensuring that the water supply remains safe and standards are maintained. |
| **Limitations & Improvement areas**: |
| Despite the promising results of the Naive Bayes model in predicting water quality, there are several limitations that must be addressed to improve future analyses. One significant limitation is the class imbalance in the dataset, where 'not safe' samples vastly outnumber 'safe' ones. This imbalance can skew the model's predictions, as evidenced by the lower precision and recall for the 'safe' class. Addressing this issue through techniques such as resampling, synthetic data generation, or adjusting class weights during model training could enhance the model's performance in identifying 'safe' water samples.  Another limitation lies in the feature set used for modeling. While the selected features are relevant, exploring additional parameters or engineering new features could provide a more comprehensive view of water quality and improve the model's predictive power. Furthermore, the current preprocessing steps, such as replacing outliers with mean values, might not be the most effective approach. Alternative methods like robust scaling or transforming the data distribution could yield better results.  In summary, future work should focus on addressing data imbalance, enhancing feature selection and engineering, and exploring advanced algorithms to build a more robust model for water quality assessment. |

|  |
| --- |
| **Appendix** |
| import pandas as pd  from sklearn.model\_selection import train\_test\_split  from sklearn.preprocessing import StandardScaler  from sklearn.naive\_bayes import GaussianNB  from sklearn.metrics import accuracy\_score, classification\_report  # Load and clean the data  data = pd.read\_csv('waterQuality1.csv')  data['ammonia'] = pd.to\_numeric(data['ammonia'], errors='coerce')  data['is\_safe'] = pd.to\_numeric(data['is\_safe'], errors='coerce')  data\_cleaned = data.dropna()  # Replace outliers with mean  def replace\_outliers\_with\_mean(df, column):  Q1 = df[column].quantile(0.25)  Q3 = df[column].quantile(0.75)  IQR = Q3 - Q1  lower\_bound = Q1 - 1.5 \* IQR  upper\_bound = Q3 + 1.5 \* IQR  mean\_value = df[column].mean()  df.loc[(df[column] < lower\_bound) | (df[column] > upper\_bound), column] = mean\_value  return df  data\_cleaned = replace\_outliers\_with\_mean(data\_cleaned, 'aluminium')  data\_cleaned = replace\_outliers\_with\_mean(data\_cleaned, 'arsenic')  # Remove outliers for 'nitrites'  Q1 = data\_cleaned['nitrites'].quantile(0.25)  Q3 = data\_cleaned['nitrites'].quantile(0.75)  IQR = Q3 - Q1  lower\_bound = Q1 - 1.5 \* IQR  upper\_bound = Q3 + 1.5 \* IQR  data\_cleaned = data\_cleaned[(data\_cleaned['nitrites'] >= lower\_bound) & (data\_cleaned['nitrites'] <= upper\_bound)]  # Select key variables  key\_variables = ['cadmium', 'arsenic', 'viruses', 'uranium', 'nitrates', 'aluminium', 'chloramine', 'is\_safe']  data\_key\_vars = data\_cleaned[key\_variables]  # Split the data into training and testing sets  X = data\_key\_vars.drop('is\_safe', axis=1)  y = data\_key\_vars['is\_safe']  X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)  # Standardize the data  scaler = StandardScaler()  X\_train\_scaled = scaler.fit\_transform(X\_train)  X\_test\_scaled = scaler.transform(X\_test)  # Train the Naive Bayes model  model = GaussianNB()  model.fit(X\_train\_scaled, y\_train)  # Predict on the test set  y\_pred = model.predict(X\_test\_scaled)  # Evaluate the model  accuracy = accuracy\_score(y\_test, y\_pred)  report = classification\_report(y\_test, y\_pred)  print(f"Model Accuracy: {accuracy}")  print(f"Classification Report:\n{report}")  # Predict probabilities  y\_prob = model.predict\_proba(X\_test\_scaled)[:, 1]  # Compute ROC curve and ROC area  fpr, tpr, thresholds = roc\_curve(y\_test, y\_prob)  roc\_auc = roc\_auc\_score(y\_test, y\_prob)  # Plot ROC curve  plt.figure()  plt.plot(fpr, tpr, color='orange', lw=2, label=f'ROC curve (area = {roc\_auc:.2f})')  plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')  plt.xlim([0.0, 1.0])  plt.ylim([0.0, 1.0])  plt.xlabel('False Positive Rate')  plt.ylabel('True Positive Rate')  plt.title('Receiver Operating Characteristic')  plt.legend(loc="lower right")  plt.show() |

**References**

 Naive Bayes Classifier. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/naive\_bayes.html

 ROC Curve. (n.d.). Retrieved from <https://en.wikipedia.org/wiki/Receiver_operating_characteristic>

 Brownlee, J. (2016, August 2). How to Calculate Precision, Recall, and F1-Score for Imbalanced Classification. Machine Learning Mastery. Retrieved from https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/

 Analytics Vidhya. (2021, October 20). Implementing Artificial Neural Network Classification in Python From Scratch. Retrieved from <https://www.analyticsvidhya.com/blog/2021/10/implementing-artificial-neural-networkclassification-in-python-from-scratch/>

Kubat, M. (2017). An Introduction to Machine Learning. Springer International Publishing. Retrieved from <https://www.dbooks.org/>