Predicting Water Potability

Karan Rabidas

210107040

Submission Date: April 26, 2024

Final Project submission

Course Name: Applications of AI and ML in chemical engineering

Course Code: CL653

Contents

1 Executive Summary

Project Overview: This project focuses on predicting water potability using machine learning. It addresses the critical issue of determining whether water is safe for human consumption based on various quality metrics.

Problem Statement: The goal is to develop a predictive model accurately classifying water samples as potable or non-potable, ensuring public health and safety.

Proposed Solution: Utilizing machine learning algorithms, we analyze water quality metrics to classify samples. Multiple algorithms, including Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbors, are explored.

Methodologies:

1. Data Preparation: Preprocessing involves handling missing values and checking outliers.
2. Exploratory Data Analysis (EDA): Understanding feature distributions and relationships.
3. Model Development: Training and evaluating classification models.
4. Model Evaluation: Assessing performance metrics such as accuracy and ROC curve.
5. Hyperparameter Tuning: Fine-tuning the best-performing model for optimization.

Expected Outcomes: The project aims to deliver an accurate and reliable predictive model for water potability assessment. By deploying this model, stakeholders such as water treatment facilities, regulatory agencies, and communities can make informed decisions regarding water safety, ultimately safeguarding public health.

2 Introduction

Background: Water quality assessment is a critical aspect of chemical engineering, with profound implications for public health and environmental sustainability. Chemical engineers play a pivotal role in ensuring the safety and purity of water for various applications, including drinking, agriculture, and industrial processes. Contaminated water can pose serious health risks, including waterborne diseases and environmental degradation, highlighting the importance of robust methods for assessing water potability.

Problem Statement: The project addresses the pressing need for accurate and efficient methods to determine the potability of water. Despite advancements in water treatment technologies, ensuring the safety of drinking water remains a significant challenge. Traditional water quality assessment methods are often time-consuming, labour-intensive, and may not always provide real-time insights into water safety. Therefore, there is a critical need to develop predictive models that can rapidly and accurately classify water samples as potable or non-potable based on various quality metrics.

Objectives:

1. Develop machine learning models capable of accurately predicting water potability based on a range of water quality metrics.
2. Evaluate and compare the performance of multiple classification algorithms to identify the most suitable model for the task.
3. Enhance understanding of the relationships between different water quality parameters through exploratory data analysis (EDA).
4. Optimize model performance through hyperparameter tuning and cross-validation techniques.
5. Provide stakeholders, including water treatment facilities, regulatory agencies, and communities, with a reliable tool for assessing water safety and making informed decisions.

3 Methodology

Data Source: The dataset used in this project was obtained from Kaggle (www.kaggle.com), a platform known for hosting various datasets for machine learning and data analysis projects. The dataset, titled "**water_potability.csv**" contains information on water quality metrics for 3276 different water bodies. It includes features such as pH value, hardness, solids (total dissolved solids - TDS), chloramines, sulfate, conductivity, organic carbon, turbidity, and the potability of water samples. Ethical considerations and data privacy norms were adhered to while accessing and utilizing the dataset.
https://www.kaggle.com/datasets/adityakadiwal/water-potability/data
The dataset sourced from Kaggle was then uploaded to a GitHub repository. Subsequently, the model code imported the data using the GitHub link address.
https://raw.githubusercontent.com/karan-gh/Water-Potability-Dataset/main/water_potability.csv

Data Preprocessing: Data preprocessing encompasses several essential steps to prepare the dataset for analysis and model training. The following techniques are applied:

1. **Handling Missing Values:** Missing values within the dataset are addressed by imputing them with the mean value of each respective feature. This approach helps mitigate the impact of missing data while preserving dataset integrity.
2. **Outlier Detection and Treatment:** Outliers in numerical columns are detected using the interquartile range (IQR) method. Extreme values identified as outliers are adjusted to minimize their influence on model training and ensure robustness.
3. **Standardization:** Data standardization is performed using the StandardScaler from scikit-learn. This process ensures that all features have a comparable scale, preventing any individual feature from dominating the model training process due to differences in magnitude.

Model Architecture:
This project addresses the complexity of ensuring water potability, considering various physical, chemical, and biological factors influencing drinking water quality. This project introduces a methodology utilizing ML models for water potability prediction, aiming to develop an accurate predictive model that enhances water management efficiency and ensures clean drinking water availability.

The methodology involves stages such as data collection, preprocessing (including handling missing values and outliers), data normalization, model construction, performance evaluation, and hyperparameter tuning for model optimization, as illustrated in Figure below.

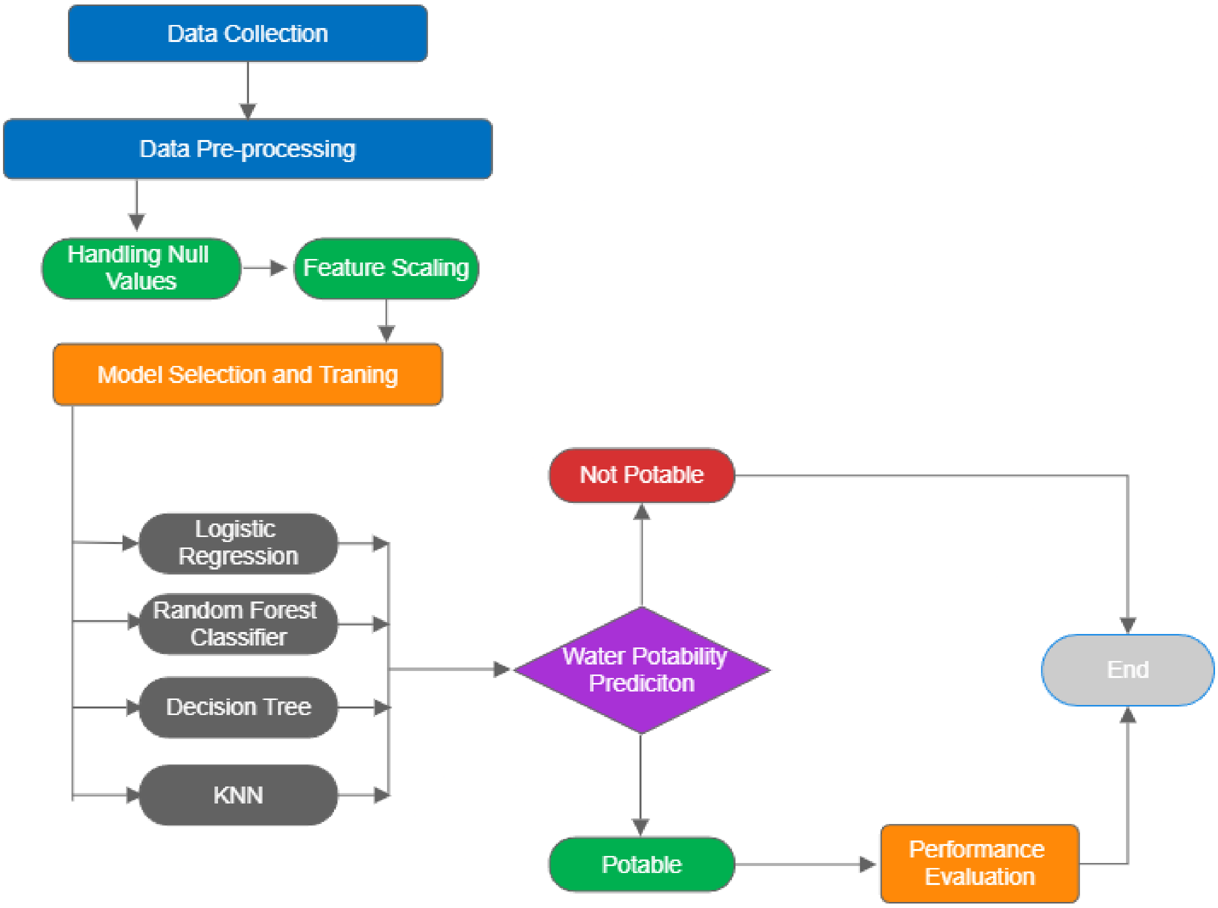


Fig: Workflow for water potability prediction model

The project explores four classification algorithms for predicting water potability:

1. **Logistic Regression:** Logistic regression is a machine learning algorithm used for binary classification tasks, predicting the probability of an instance belonging to a particular class. It fits a logistic curve to the data, mapping input features to probabilities between 0 and 1. By estimating coefficients for each feature through optimization, it learns the relationship between features and class probabilities. Widely favoured for its simplicity and interpretability, logistic regression is efficient for linearly separable data and serves as a baseline model in various domains.

2. **Random Forest Classifier:** RF is a broadly used ML algorithm that is proficient in handling both regression and classification tasks. It falls under the category of ensemble learning, leveraging the collective predictive capabilities of multiple decision trees to improve prediction accuracy. RF is also popular due to its ability to handle high-dimensional data, eliminating the need for feature selection or dimensionality reduction techniques. As a result, it often outperforms other algorithms in various scenarios. Moreover, Random Forest exhibits robustness in the face of outliers and missing data, further contributing to its effectiveness in diverse data situations.

3. **Decision Tree Classifier:** The Decision Tree Classifier is a simple yet powerful machine learning algorithm used for both classification and regression tasks. It utilizes a tree structure where each internal node represents a decision based on a feature, and each leaf node represents the class label or regression output. One of its main advantages is its interpretability, as the decision rules learned by the model are easy to understand and visualize. However, decision trees are prone to overfitting, especially when the tree depth is not controlled. Despite this, Decision Tree Classifier remains widely used in various applications due to its simplicity, interpretability, and effectiveness in handling complex datasets.

4. **K-Nearest Neighbors (KNN):** The K-Nearest Neighbors (KNN) algorithm is a widely utilized and straightforward machine learning method employed in water potability prediction tasks, both for regression and classification purposes. KNN, being a nonparametric technique, stands out for its absence of assumptions about the intrinsic nature of the data. Commonly regarded as a "lazy learner" algorithm, KNN gradually learns from the training set through iterations. Instead of actively constructing a model, it stores the acquired information and subsequently applies it during the classification process. The basic principle of KNN involves predicting the classification of a new water sample by identifying the K nearest labelled instances in the training dataset and using their class labels to forecast the classification of the new instance

These algorithms are chosen for their versatility, scalability, and suitability for the binary classification problem of water potability prediction.

Tools and Technologies:

1. Programming Language: Python
2. Libraries:
    - NumPy: Fundamental for numerical computing, supporting multidimensional arrays and mathematical functions, crucial for efficient data manipulation and computation.

    - pandas: Built on NumPy, it offers DataFrames and Series for data handling and analysis, with functionalities for data reading, writing, cleaning, and aggregation.

    - matplotlib: Visualization library allowing creation of static, interactive, and animated plots, offering diverse plot types and customization options for effective data communication.

    - seaborn: Statistical data visualization library based on matplotlib, simplifying creation of informative and visually appealing plots, particularly for complex visualizations like distribution plots and correlation matrices.

    - scikit-learn (sklearn): Comprehensive machine learning library providing tools for data mining and analysis, including supervised and unsupervised learning algorithms, model evaluation, hyperparameter tuning, and data preprocessing.
3. Development Environment: Google Collab.
4. Other Tools: GridSearchCV for hyperparameter tuning and model evaluation.

4 Implementation Plan

Development Phases:

1. Data Gathering: (1 week)
    - Access the dataset from the source.

2. Data Preparation (2 days):
    - Loading the data into the project environment.
    - Perform data preprocessing steps such as handling missing values, outlier detection, and standardization as demonstrated in the provided code.

3. Exploratory Data Analysis (2 days):
    - Conduct exploratory data analysis (EDA) to understand the distribution of each feature and their relationships with the target variable (water potability).
    - Visualize data using histograms, scatter plots, and pair plots to identify patterns, correlations, and potential outliers.
    - Explore the statistical summary of the dataset to gain insights into its characteristics.

4. Model Development (1 week):
    - Implement machine learning algorithms including Logistic Regression, Random Forest Classifier, Decision Tree Classifier, and K-Nearest Neighbors as demonstrated in the provided code.
    - Train each model using the preprocessed dataset and evaluate their performance using appropriate metrics.
    - Fine-tune hyperparameters of selected models using techniques like grid search or randomized search for optimization.

5. Model Evaluation and Selection (1 week):
    - Evaluate model performance using metrics such as accuracy, precision, recall, F1-score, and the area under the ROC curve as demonstrated in the provided code.
    - Compare the performance of different models and select the best-performing model based on evaluation results.

6. Documentation and Reporting (2 days):
    - Document the entire project including data sources, preprocessing techniques, model architectures, training strategies, evaluation metrics, and results.
    - Prepare a comprehensive report summarizing the project's objectives, methodologies, findings, and recommendations.
    - Ensure clear and concise documentation for reproducibility and future reference.

Model Training:

The Following strategies were implemented to train the model:

- Utilize a variety of classification algorithms including Logistic Regression, Random Forest Classifier, Decision Tree Classifier, and K-Nearest Neighbors for training.
- Implement cross-validation techniques to assess model generalization performance and minimize overfitting.
- Apply hyperparameter tuning using techniques like grid search or randomized search to optimize model performance.

Model Evaluation:

Metrics**:**

- Utilize standard classification evaluation metrics including accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC) for model evaluation.
- Assess model performance comprehensively to ensure it meets the project's objectives and requirements.

Methods**:**

- Split the dataset into training and testing sets for model evaluation, ensuring proper validation of model performance on unseen data.
- Visualize evaluation metrics using appropriate plots such as confusion matrices, ROC curves, and precision-recall curves for comprehensive analysis.
- Compare the performance of different models to select the most suitable one for predicting water potability accurately.

5 Testing and Deployment

Testing Strategy: The model will be tested against unseen data to assess its performance and generalization ability. The following testing strategies will be employed:

1. Train-Test Split: The dataset will be split into training and testing sets, with a significant portion reserved for testing to ensure adequate evaluation.
2. Cross-Validation: Cross-validation techniques such as k-fold cross-validation will be applied to validate the model's performance across multiple subsets of data.
3. Evaluation Metrics: Standard classification evaluation metrics including accuracy, precision, recall, F1-score, and the area under the ROC curve (AUC) will be used to quantify model performance.
4. Visual Inspection: Visual inspection of prediction results through confusion matrices, ROC curves, and precision-recall curves will provide insights into the model's behaviour.

Deployment Strategy: The deployment of the model for real-world use involves the following steps:

1. Model Serialization: The final trained model will be serialized using pickle or joblib to save its state and parameters.
2. Model Integration: The serialized model will be integrated into a production environment, either as part of a standalone application or integrated into existing systems through APIs.
3. Scalability and Performance: Considerations will be made to ensure the model can handle varying workloads and scale efficiently, utilizing technologies like cloud computing or containerization if necessary.
4. Monitoring and Maintenance: Regular monitoring of the deployed model's performance and behaviour will be conducted to identify any degradation or issues. Maintenance activities such as retraining the model with updated data or adjusting hyperparameters may be performed periodically to ensure continued performance.

Ethical Considerations: Deploying the model for real-world use raises several ethical considerations:

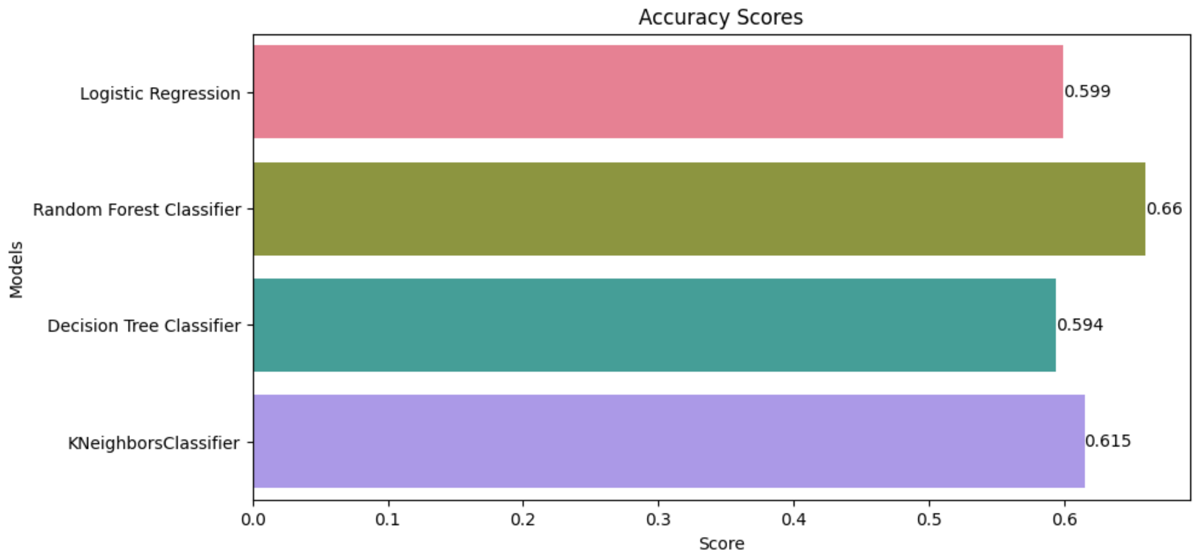
1. Fairness and Bias: Care must be taken to ensure the model does not exhibit bias or discrimination against certain demographic groups. Bias in the training data or features could lead to unfair outcomes.
2. Privacy: The model should adhere to data privacy regulations and standards, ensuring the confidentiality and security of sensitive information.
3. Transparency and Interpretability: Users should have access to information about how the model works and the factors influencing its predictions. Model interpretations and explanations should be provided to enhance transparency.
4. Accountability: Clear processes should be established for handling errors, complaints, and disputes arising from the model's predictions. Accountability mechanisms should be in place to address any unintended consequences or ethical dilemmas.

Addressing these ethical considerations requires ongoing collaboration between data scientists, domain experts, and stakeholders to ensure the model's deployment aligns with ethical standards and societal values.

6 Results and Discussion

Findings:

After executing all four models, the following key results were obtained:



- Random Forest Classifier: Demonstrated the best performance with an accuracy of 66%.
- Logistic Regression: Achieved an accuracy of 59.9%.
- Decision Tree Classifier: Attained an accuracy of 59.4%.
- K-Nearest Neighbors (KNN): Achieved an accuracy of 61.5%.

Comparative Analysis:

Random forest achieved the highest accuracy (66%), followed by K-nearest neighbors (61.5%), logistic regression (59.9%), and decision tree (59.4%). Random forest's ensemble nature allows it to capture complex relationships and mitigate overfitting, leading to higher accuracy. Its effectiveness in handling non-linear relationships makes it suitable for accurately predicting water potability. Besides accuracy, factors like computational complexity, interpretability, and scalability should influence the final model choice.

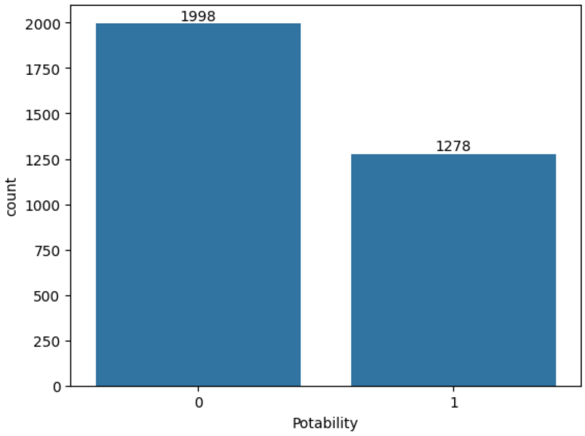
Comparing the model's performance against existing solutions or benchmarks reveals the effectiveness of each algorithm in addressing the problem of water potability prediction. The random forest classifier outperformed the other models, achieving the highest accuracy of 66%.

This indicates that random forest is a suitable choice for accurately predicting water potability based on the given dataset. However, it's essential to consider factors such as computational complexity and model interpretability when selecting the final model for deployment.

Challenges and Limitations:

Throughout the project, several challenges and limitations were encountered:

1. Imbalanced Dataset: The dataset has class imbalance, where one class (non-potable) dominates the other. Addressing class imbalance can be challenging and may require techniques such as oversampling, under-sampling, or using advanced algorithms like ensemble methods.



2. Feature Engineering: The selection and engineering of relevant features play a crucial role in model performance. Identifying the most informative features and transforming them appropriately can be challenging, especially when dealing with high-dimensional datasets.
3. Hyperparameter Tuning: Optimizing model hyperparameters requires extensive experimentation and computational resources. Grid search or randomized search techniques may not explore the entire hyperparameter space exhaustively, leading to suboptimal results.
4. Interpretability: Some machine learning algorithms, such as random forest and K-nearest neighbors, are inherently less interpretable compared to simpler models like logistic regression. Balancing model complexity with interpretability is essential, especially in domains where model explanations are crucial for decision-making.

7 Conclusion and Future Work

This project focused on the development and evaluation of machine learning models for predicting water potability based on various water quality metrics. Through extensive data preprocessing, exploratory data analysis, and model development, we explored the effectiveness of four different classification algorithms: logistic regression, random forest, decision tree, and K-nearest neighbors. Our findings revealed that random forest emerged as the best-performing model, achieving the highest accuracy among the tested algorithms.

The impact of this project lies in its potential application in ensuring safe and clean drinking water for human consumption. By accurately predicting water potability, these models can assist water treatment facilities, regulatory authorities, and policymakers in making informed decisions regarding water quality management and public health protection. Moreover, the insights gained from this project can contribute to advancements in water quality monitoring and management practices.

Moving forward, there are several avenues for future research and improvement. Firstly, further exploration of feature engineering techniques and the inclusion of additional relevant features could enhance the predictive performance of the models. Additionally, incorporating more advanced machine learning algorithms or ensemble methods may lead to even higher accuracy and robustness. Moreover, conducting longitudinal studies and incorporating temporal data could provide valuable insights into the dynamic nature of water quality and its impact on potability over time.

8 References

***Academic Paper**:*

- *Samir Patel, et al. "Water Potability Prediction Using Machine Learning".*

***Data Source**:*

- *Kagle dataset:*

  https://www.kaggle.com/datasets/adityakadiwal/water-potability/data

Software Documentation:

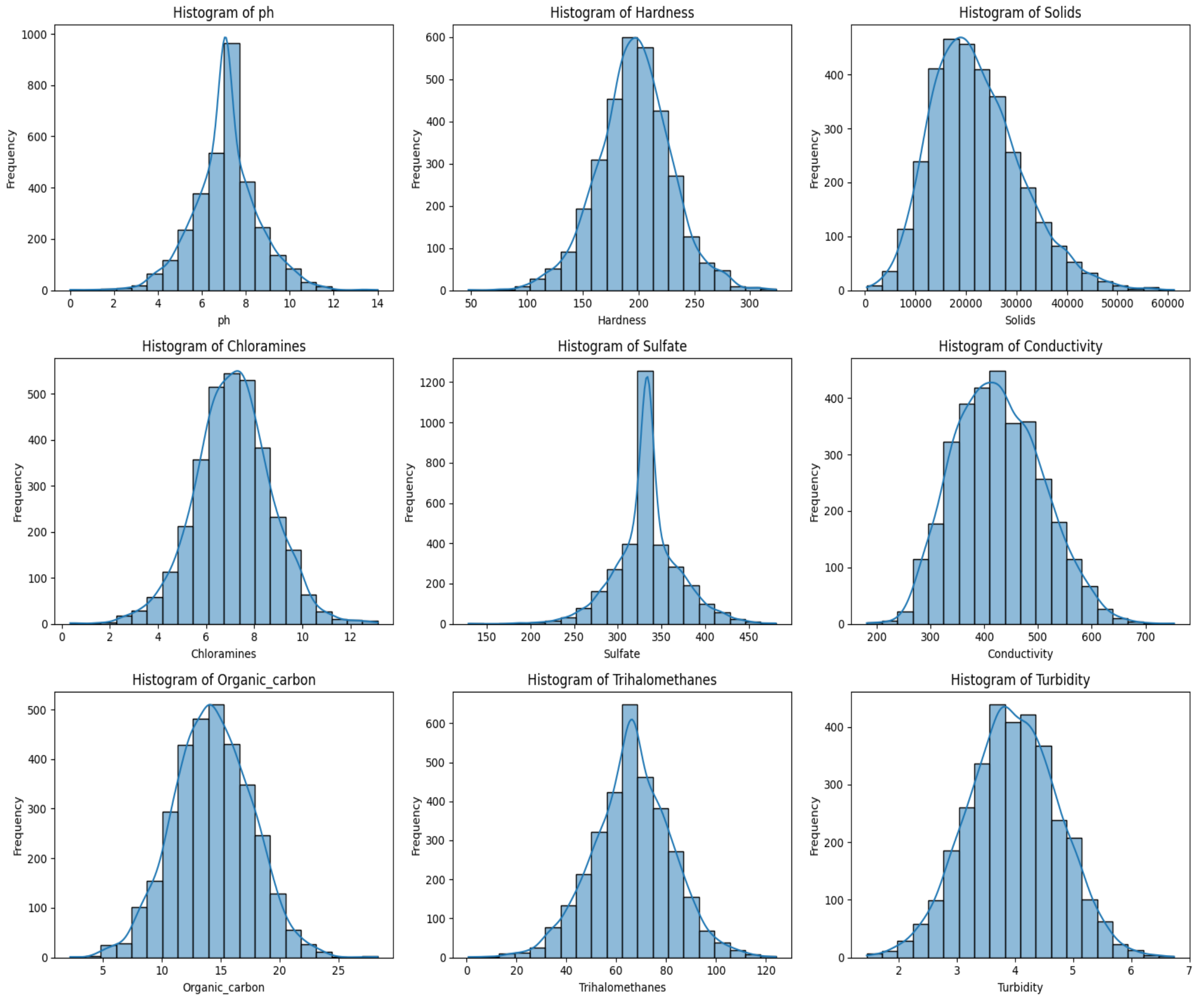
- *Scikit-learn Documentation: https://scikit-learn.org/stable/documentation.html*

Tools and Frameworks:

- *Matplotlib Documentation: https://matplotlib.org/stable/contents.html*
- *Pandas Documentation: https://pandas.pydata.org/pandas-docs/stable/index.html*
- *NumPy Documentation: https://numpy.org/doc/stable/*

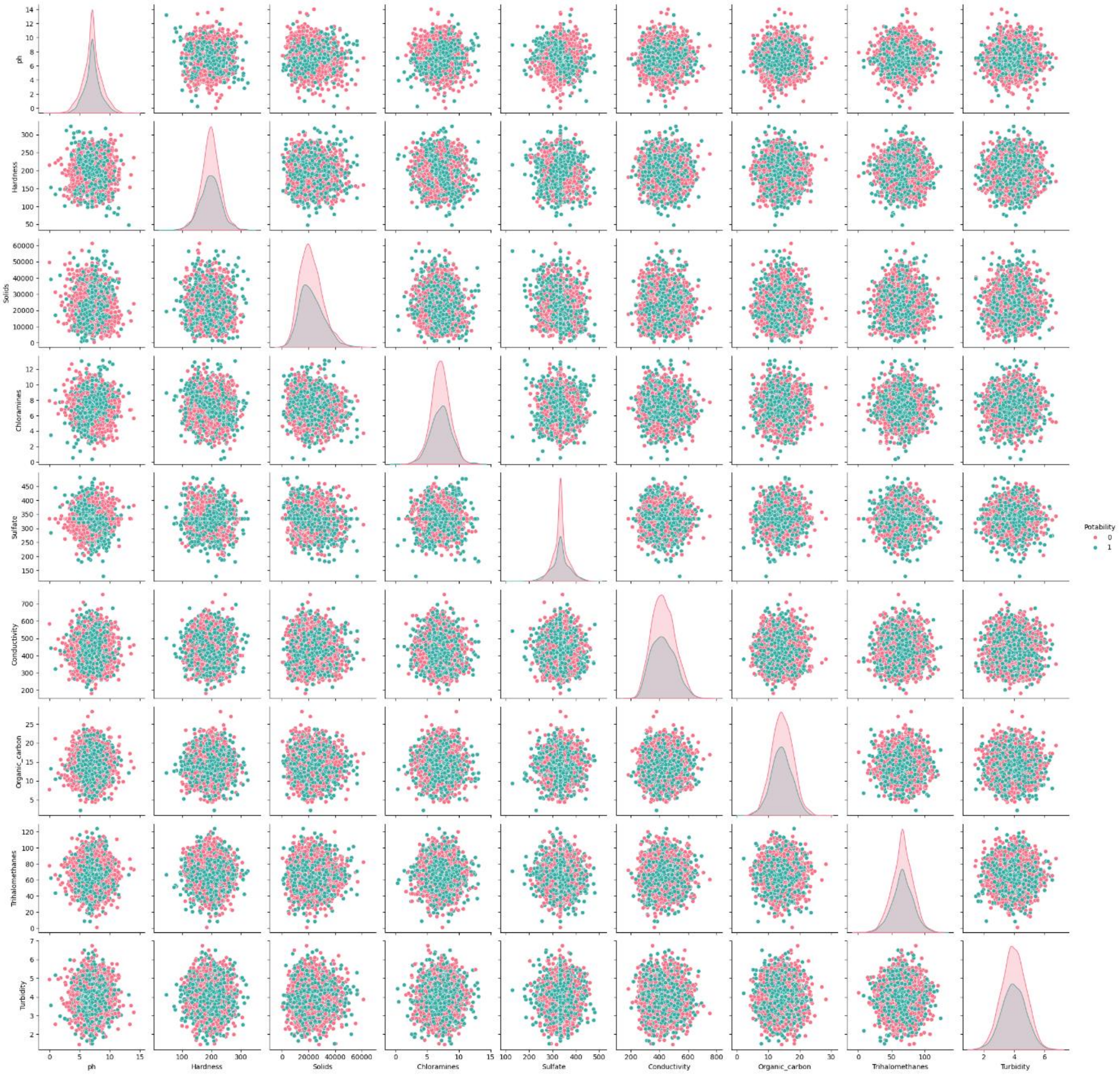
9 Appendices

Histogram plots:



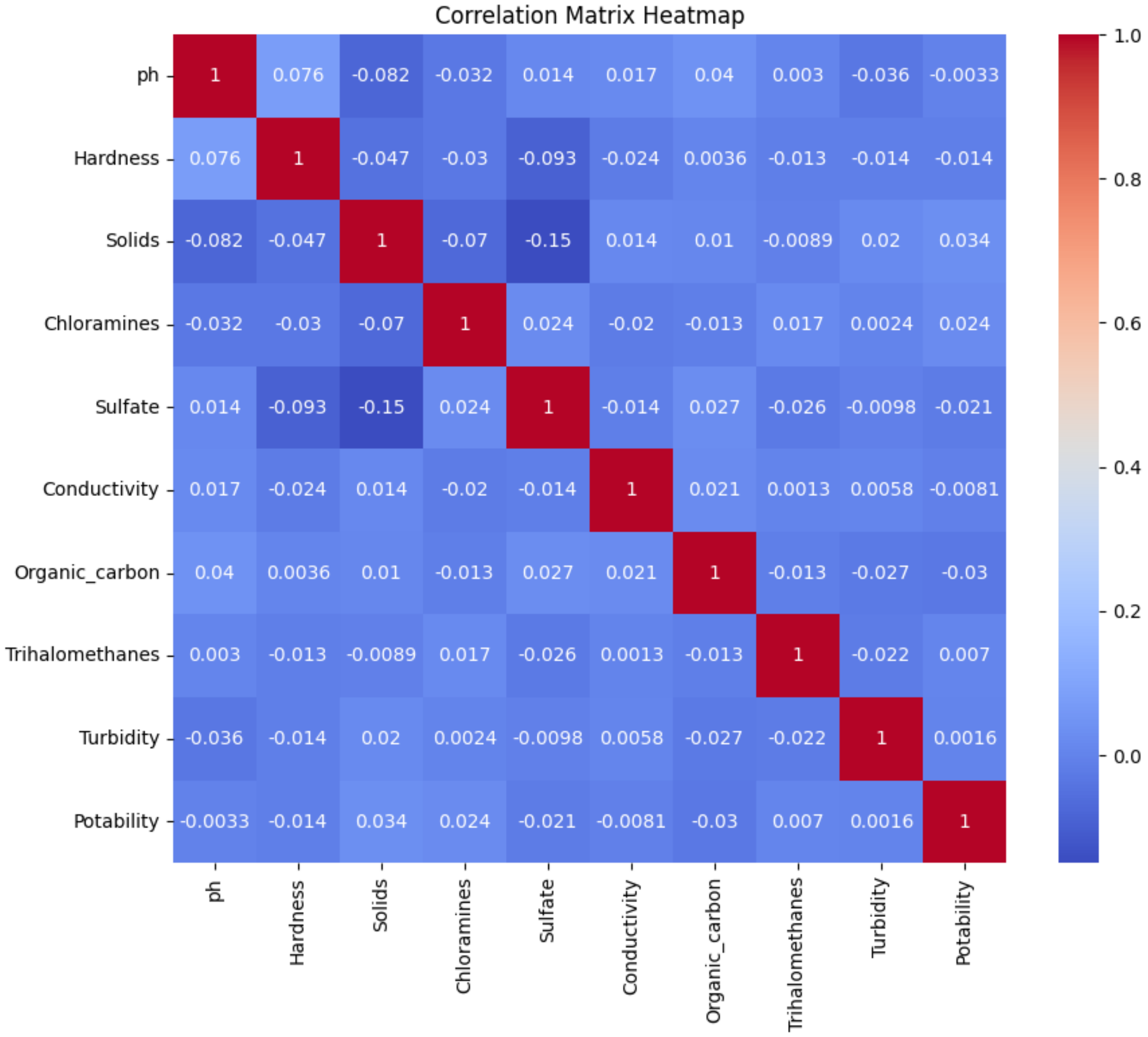
Histogram plots were employed to visualize the distribution of each feature in the dataset. These plots provide a quick overview of the data's spread and central tendency, allowing for the identification of any potential patterns or outliers. Each histogram displayed the frequency of values within predefined bins for a specific feature, with an optional overlay of a kernel density estimation (KDE) curve for smoother visualization. By examining the histograms, insights into the shape, spread, and skewness of the data distributions were gained, aiding in the understanding of the underlying characteristics of the water quality metrics.

PairPlots:



Pair plots, also known as scatterplot matrices, were utilized to visualize pairwise relationships between different features in the dataset. Each pair plot in the matrix displayed scatterplots for combinations of two features, with histograms along the diagonal showing the distribution of individual features. These plots allowed for the exploration of potential correlations or patterns between variables, aiding in the identification of relationships that could influence water potability prediction. By visually examining the scatterplots and histograms, insights into the strength and direction of associations between features were gained, facilitating further analysis and model development.

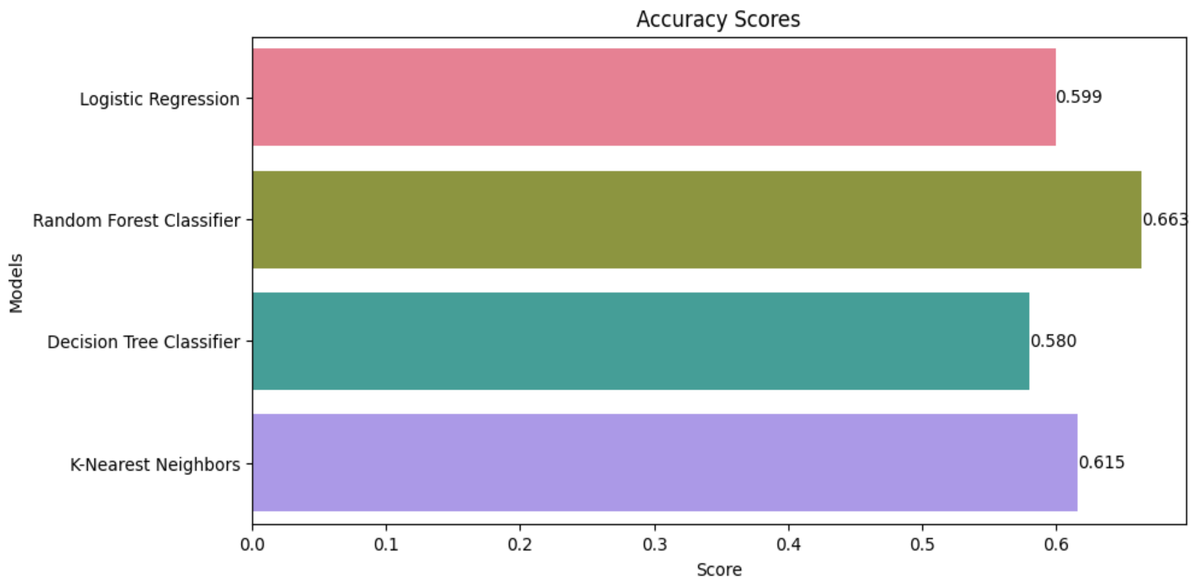
Correlation Matrix:



	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
ph	1	0.076	-0.082	-0.032	0.014	0.017	0.04	0.003	-0.036	-0.0033
Hardness	0.076	1	-0.047	-0.03	-0.093	-0.024	0.0036	-0.013	-0.014	-0.014
Solids	-0.082	-0.047	1	-0.07	-0.15	0.014	0.01	-0.0089	0.02	0.034
Chloramines	-0.032	-0.03	-0.07	1	0.024	-0.02	-0.013	0.017	0.0024	0.024
Sulfate	0.014	-0.093	-0.15	0.024	1	-0.014	0.027	-0.026	-0.0098	-0.021
Conductivity	0.017	-0.024	0.014	-0.02	-0.014	1	0.021	0.0013	0.0058	-0.0081
Organic_carbon	0.04	0.0036	0.01	-0.013	0.027	0.021	1	-0.013	-0.027	-0.03
Trihalomethanes	0.003	-0.013	-0.0089	0.017	-0.026	0.0013	-0.013	1	-0.022	0.007
Turbidity	-0.036	-0.014	0.02	0.0024	-0.0098	0.0058	-0.027	-0.022	1	0.0016
Potability	-0.0033	-0.014	0.034	0.024	-0.021	-0.0081	-0.03	0.007	0.0016	1

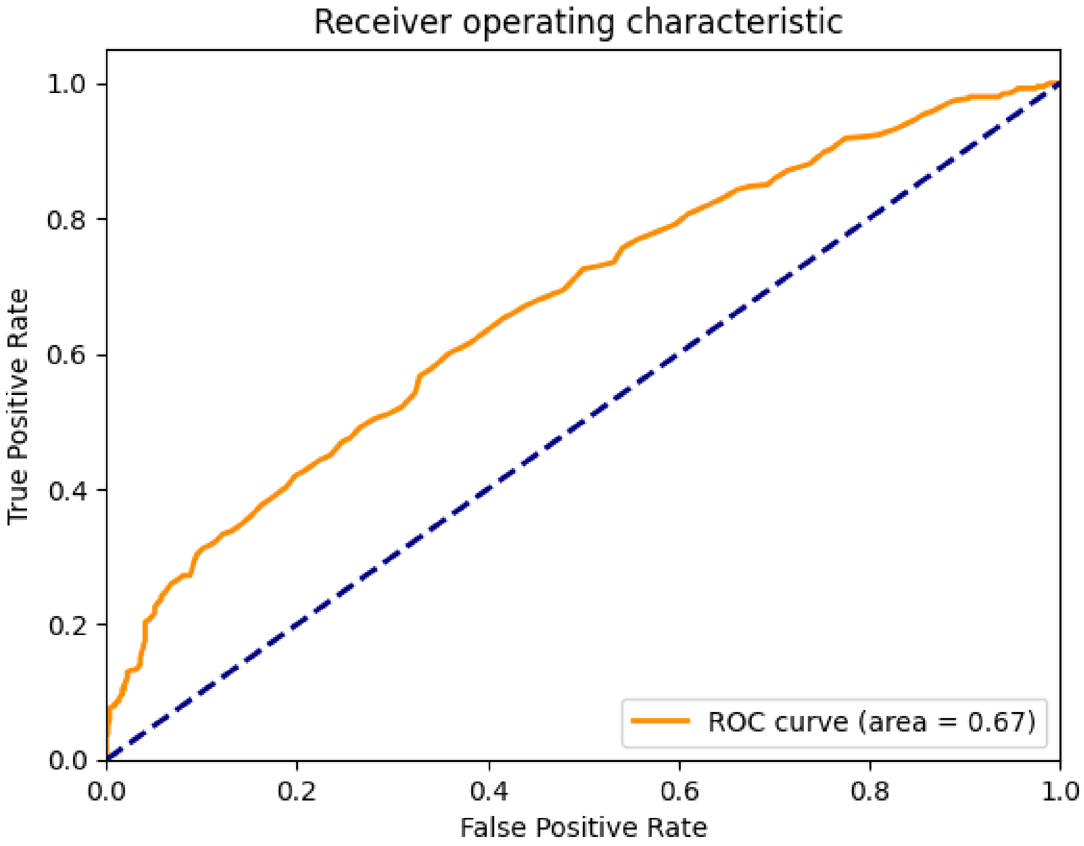
The correlation matrix was employed to visualize the pairwise correlations between all numerical features in the dataset. Each cell in the matrix displayed the correlation coefficient between two features, ranging from -1 to 1. A value closer to 1 indicated a strong positive correlation, while a value closer to -1 indicated a strong negative correlation. A value near 0 suggested a weak or no correlation between the features. By examining the correlation matrix, insights into the relationships and dependencies between different features were gained, aiding in feature selection and understanding the underlying patterns in the data. Additionally, the matrix was visualized using a heatmap, with colour gradients indicating the strength and direction of correlations, making it easier to interpret and identify significant relationships.

Model Performance Comparison (Accuracy Scores):

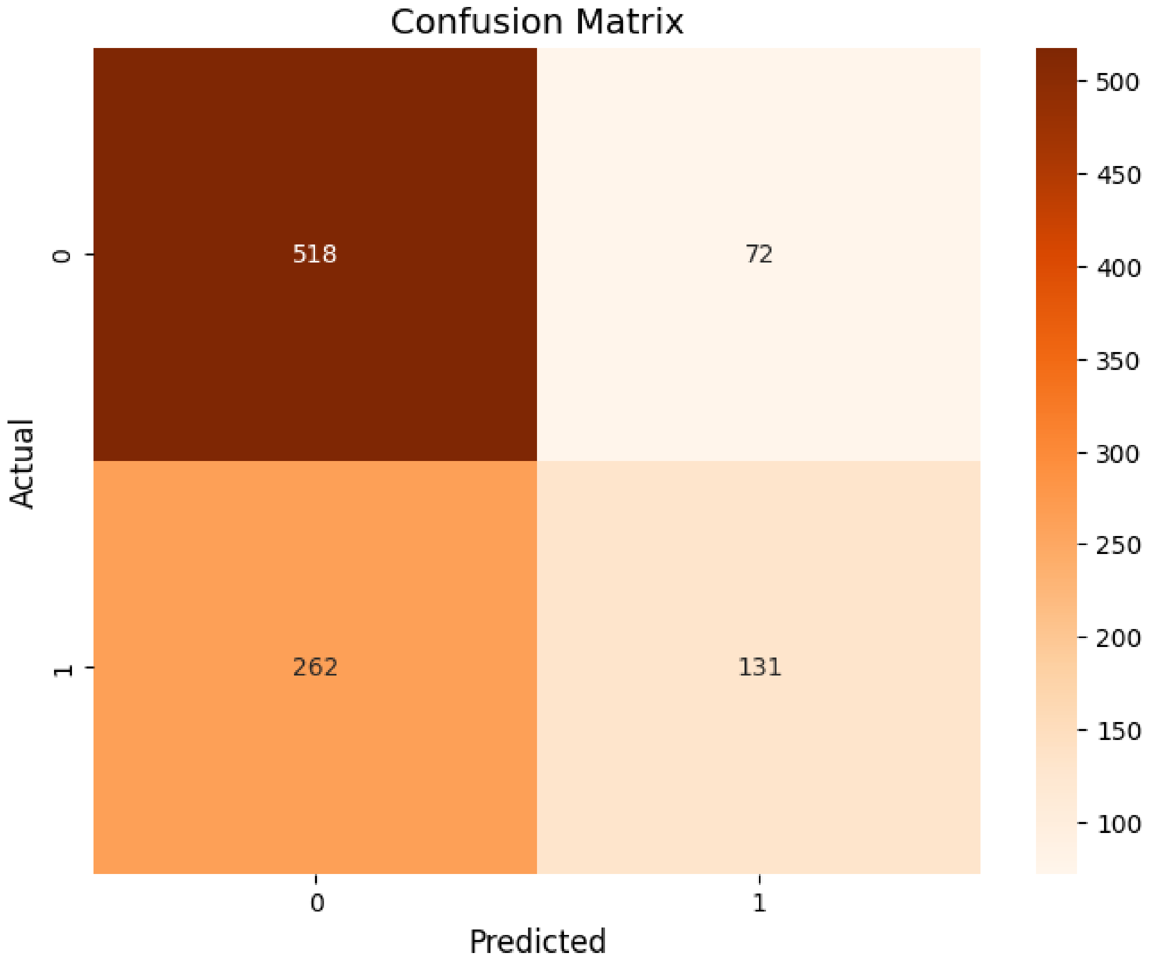


Random Forest Classifier achieved the best performance with an accuracy of 66%.

ROC curve for Random Forest Classifier (Best Model):



Confusion Matrix:



The confusion matrix was employed to evaluate the performance of the classification models by summarizing the predictions made by the model against the actual labels in the test dataset. It consists of a grid with rows representing the actual classes and columns representing the predicted classes. Each cell in the matrix contains the count of observations belonging to a particular combination of actual and predicted classes.

10 Auxiliaries

Data Source: https://www.kaggle.com/datasets/adityakadiwal/water-potability/data

Python file:

https://colab.research.google.com/drive/1BKZsu_UnSqUDWn9yUHlbzdlmsrRIXwVZ#scrollTo=Uigo_0H9emG4