**WATER QUALITY PREDICTION**

# Bachelor of Technology

in

# Computer Science & Engineering

## By

**2203A51164 V. Rahul**

**Under the Guidance of**

# D.Ramesh

# Assistant Professor, Department of CSE.

**Submitted to**



# DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

## SRUNIVERSITY, ANANTHASAGAR, WARANGAL





**DEPARTMENT OF COMPUTER SCIENCE**

**&**

**ENGINEERING**

**CCCERTIFICATE**

This is to certify that the Project Report entitled “**WATER QUALITY**

**PREDICTION**” is a record of Bonafide work carried out by V. Rahul , bearing Roll No 2203A51164 during the academic year 2022-2023 in partial fulfillment of the award of the degree of Bachelor of Technology in Computer Science Engineering by the SR UNIVERSITY, WARANGAL.

# Supervisor Head of the Department

Mr. D. Ramesh Dr. M. Sheshikala

Asst. Professor, Assoc. Prof .& HOD (CSE)

SR University SR University

**ACKNOWLEDGEMENT**

I express our thanks to course coordinator Mr. D. Ramesh, Asst. prof. for guiding us from the beginning through the end of the course project. We express our gratitude to head of the department CS&AI, Dr. M.Sheshikala,Associate Professor for encouragement, support and insightful suggestions. We truly value their consistent feedback on our progress, which was always constructive and encouraging and ultimately drove us to the right direction.

We wish to take this opportunity to express our sincere gratitude and deep sense of respect to our beloved Dean, School of Computer Science and Artificial Intelligence, Dr C. V. Guru Rao, for his continuous support and guidance to complete this project in the institute.

Finally, we express our thanks to all teaching and non-teaching staff of the department for their suggestions and timely support.

# ABSTRACT

Water quality prediction is crucial for ensuring the safety and sustainability of water resources, as it enables proactive measures to mitigate potential risks to public health and the environment. In this project, we propose a comprehensive approach to water quality prediction leveraging machine learning techniques. The primary objective is to develop accurate and reliable models capable of predicting water quality parameters based on various environmental factors.

First, we gather information about different things in the water like ph, Hardness, Solids, and by analyzing the parameters we can determine the ‘Potability’ of the water.

The proposed approach offers several benefits, including enhanced water quality monitoring, early detection of pollution events, improved resource allocation, and better management of water treatment processes.

The goal is to make it easier for people to know if their water is safe to drink. By using computers to help, we hope to make sure that everyone has access to clean and safe drinking water. This project is important because it helps protect people's health and makes sure everyone has the water they need to stay well.

# Table of Contents

|  |  |  |
| --- | --- | --- |
| **S.NO** | **Content** | **PageNo** |
| 1 | Introduction | 1 |
| 2 | Literature Review | 2 |
| 3 | Methodology | 4 |
| 4 | Data Pre-processing | 7 |
| 5 | Results | 11 |
| 6 | Conclusion | 20 |
| 7 | Future scope | 20 |
| 8 | References | 20 |

**1. INTRODUCTION:**

# Having clean water to drink is really important for keeping us healthy. But sometimes, it's hard to know if the water is clean or if it might make us sick. That's where this project comes in. We're using AI to help us figure out if water is clean and safe to drink.

# The goal of this project is to make it easier for people to know if the water they're drinking is safe. We're doing this by training AI models and predicting the quality of the water by considering some of the factors that helps to predict the water quality.

# Here's how it works: first, we gather a lot of information about the water like

# ph, Hardness, Solids and organic carbon.

# We train the model with some sample data and the trained model can predict the water quality by comparing the given data with the trained data.

# By using AI models to help us predict water quality, we hope to make sure that everyone has access to clean and safe drinking water. This project is really important because it helps protect people's health and ensures that everyone can have the water they need to stay well.

# 2. LITERATURE REVIEW:

This research explores the methodologies that have been employed to help solve problems related to water quality. Typically, conventional lab analysis and statistical analysis are used in research to aid in determining water quality, while some analyses employ machine learning methodologies to assist in finding an optimized solution for the water quality problem.

Local research employing lab analysis helped us gain a greater insight into the water quality problem in Pakistan. In one such research study, Daud et al. gathered water samples from different areas of Pakistan and tested them against different parameters using a manual lab analysis and found a high presence of *E. coli* and fecal coliform due to industrial and sewerage waste. Alamgir et al. tested 46 different samples from Orangi town, Karachi, using manual lab analysis and found them to be high in sulphates and total fecal coliform count.s

After getting familiar with the water quality research concerning Pakistan, we explored research employing machine learning methodologies in the realm of water quality. When it comes to estimating water quality using machine learning, Shafi et al. estimated water quality using classical machine learning algorithms namely, Support Vector Machines (SVM), Neural Networks (NN), Deep Neural Networks (Deep NN) and k Nearest Neighbors (kNN), with the highest accuracy of 93% with Deep NN. The estimated water quality in their work is based on only three parameters: turbidity, temperature and pH, which are tested according to World Health Organization (WHO) standards (Available online at URL [**https://www.who.int/airpollution/guidelines/en/**](https://www.who.int/airpollution/guidelines/en/)). Using only three parameters and comparing them to standardized values is quite a limitation when predicting water quality. Ahmad et al. employed single feed forward neural networks and a combination of multiple neural networks to estimate the WQI. They used 25 water quality parameters as the input. Using a combination of backward elimination and forward selection selective combination methods, they achieved an R2 and MSE of 0.9270, 0.9390 and 0.1200, 0.1158, respectively.

The use of 25 parameters makes their solution a little immoderate in terms of an inexpensive real time system, given the price of the parameter sensors. Sakizadeh predicted the WQI using 16 water quality parameters and ANN with Bayesian regularization. His study yielded correlation coefficients between the observed and predicted values of 0.94 and 0.77, respectively. Abyaneh predicted the chemical oxygen demand (COD) and the biochemical oxygen demand (BOD) using two conventional machine learning methodologies namely, ANN and multivariate linear regression. They used four parameters, namely pH, temperature, total suspended solids (TSS) and total suspended (TS) to predict the COD and BOD. Ali and Qamar used the unsupervised technique of the average linkage (within groups) method of hierarchical clustering to classify samples into water quality classes.

However, they ignored the major parameters associated with WQI during the learning process and they did not use any standardized water quality index to evaluate their predictions. Gazzaz et al.used ANN to predict the WQI with a model explaining almost 99.5% of variation in the data.

They used 23 parameters to predict the WQI, which turns out to be quite expensive if one is to use it for an IoT system, given the prices of the sensors. Rankovic et al. predicted the dissolved oxygen (DO) using a feedforward neural network (FNN). They used 10 parameters to predict the DO, which again defeats the purpose if it has to be used for a real-time WQI estimation with an IoT system.

# 3. METHODOLOGY:

After Data pre-processing and data visualization the next step is to apply the models on the dataset. Our dataset comes under supervised learning as it contains the labeled data (target variables, feature variables). First the dataset is splitted into training set and testing set. Then the model is trained on training set and then tested on testing set.

**3.1 Logistic Regression Algorithm:**

Logistic Regression is a statistical model that predicts the probability of a binary outcome based on one or more input variables. It is widely used in various fields, such as identifying spam emails, diagnosing diseases, and predicting customer churn, where the goal is to classify data into two categories.

* from sklearn.linear\_model import LogisticRegression
* from sklearn.metrics import confusion\_matrix, accuracy\_score, classification\_report
* model\_lg = LogisticRegression(max\_iter=120,random\_state=0, n\_jobs=20)

**3.2 K-Nearest Neighbor Algorithm:**

The K-Nearest Neighbors (KNN) algorithm is a fundamental supervised machine learning method used for both classification and regression tasks. It is a non-parametric algorithm that does not make underlying assumptions about the data distribution, making it versatile and widely applicable. KNN works by finding the K nearest neighbors to a given data point based on a distance metric, such as Euclidean distance. The class or value of the data point is then determined by the majority vote or average of the K neighbors, allowing the algorithm to adapt to different patterns and make predictions based on the local structure of the data.

# from sklearn.neighbors import KNeighborsClassifier

# model\_kn = KNeighborsClassifier(n\_neighbors=9, leaf\_size=20)

# model\_kn.fit(X\_train, y\_train)

# 3.3 Decision Tree Algorithm:

The Decision Tree Algorithm is a versatile supervised machine learning technique used for both classification and regression tasks. It constructs a tree-like structure where each internal node represents a test on an attribute, branches denote the outcomes of the test, and leaf nodes hold class labels. The algorithm recursively splits the training data into subsets based on attribute values until a stopping criterion is met, such as maximum tree depth or minimum samples required to split a node. Decision Trees select the best attribute for splitting based on metrics like entropy or Gini impurity to maximize information gain or reduce impurity after the split.

# from sklearn.tree import DecisionTreeClassifier

# model\_dt = DecisionTreeClassifier( max\_depth=4, random\_state=42)

# model\_dt.fit(X\_train,y\_train)

# 3.4 Support Vector Machine Algorithm:

Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for classification and regression tasks. It aims to find the optimal hyperplane in an N-dimensional space that separates data points of different classes with the maximum margin. SVM is versatile, efficient in high-dimensional cases, and can handle linear and nonlinear classification problems. By utilizing support vectors and kernel functions, SVM can transform data into higher-dimensional spaces for better separation. It is widely used in various applications like text classification, image recognition, and anomaly detection due to its adaptability and effectiveness in finding the best decision boundary.

* from sklearn.svm import SVC, LinearSVC
* model\_svm = SVC(kernel='rbf', random\_state = 42)
* model\_svm.fit(X\_train, y\_train)

**3.5 Random Forest Algorithm:**

Random Forest is a supervised machine learning algorithm that uses ensemble learning, where multiple decision trees are built on various subsets of the dataset. It then aggregates the results from each tree to make the final prediction based on majority voting or averaging, depending on the task (classification or regression).

* from sklearn.ensemble import RandomForestClassifier
* model\_rf = RandomForestClassifier(n\_estimators=300,min\_samples\_leaf=0.16, random\_state=42)
* model\_rf.fit(X\_train, y\_train)
* pred\_rf = model\_rf.predict(X\_test)

**3.5 Adaboost Algorithm:**

AdaBoost, short for Adaptive Boosting, is a powerful ensemble machine learning algorithm that combines multiple weak learners, often decision trees, into a strong learner. It is a supervised learning algorithm used for classification and regression tasks. AdaBoost works by sequentially training a series of weak learners on the dataset, where each subsequent learner focuses more on the instances that were misclassified by the previous learners. By assigning weights to the training instances based on their classification accuracy, AdaBoost aims to improve the overall performance of the model.

* from sklearn.ensemble import AdaBoostClassifier
* from sklearn.metrics import accuracy\_score, classification\_report, confusion\_matrix
* import seaborn as sns
* import numpy as np

# 4. DATASET PREPROCESSING:

# DATASET DESCRIPTION

# Attributes:

# Ph

# Hardness

# Solids

# Chloramines

# Sulfate

# Conductivity

# Organic Carbon

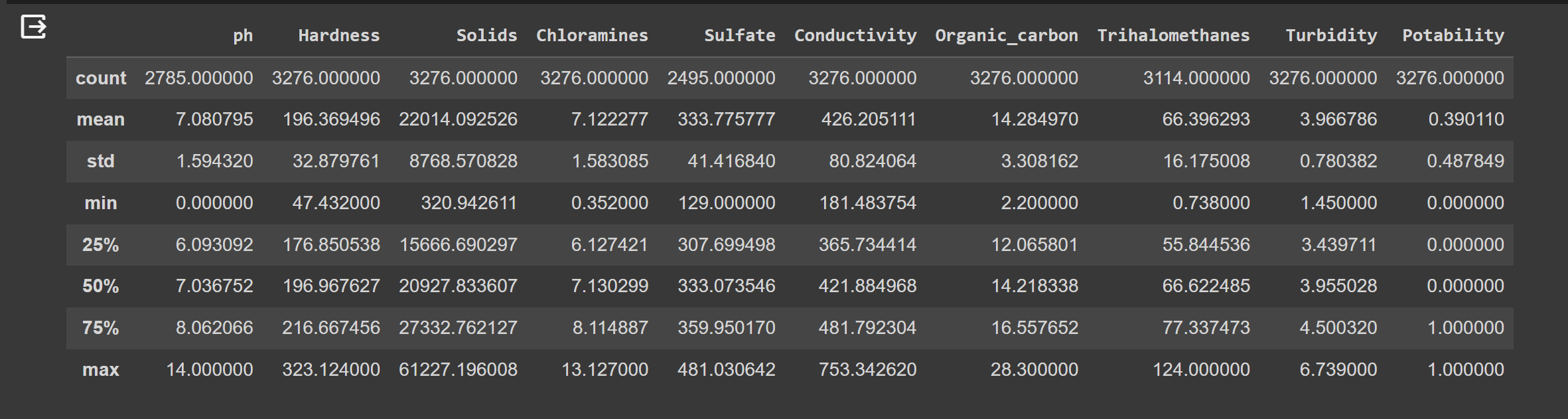
# Trihalomethanes

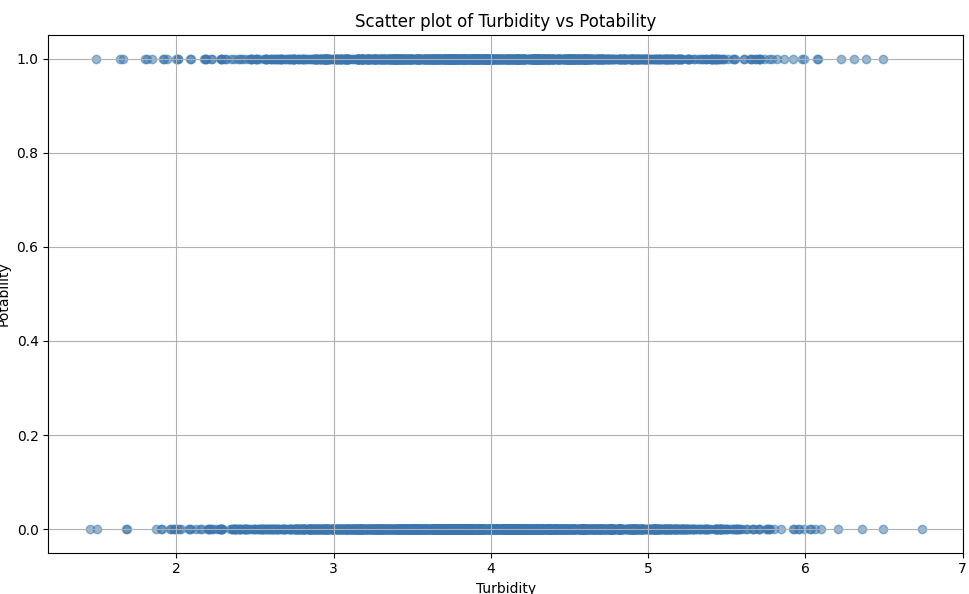
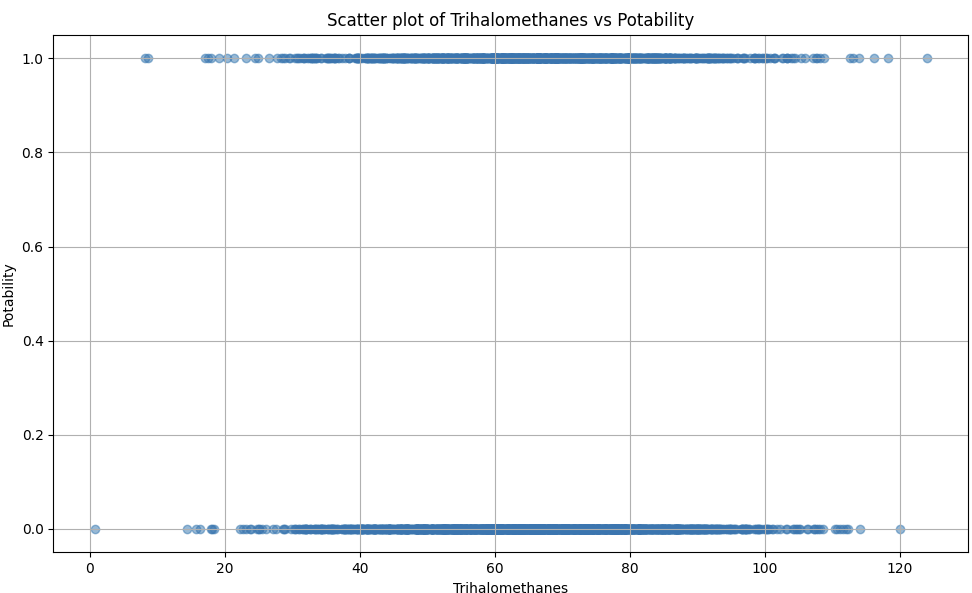
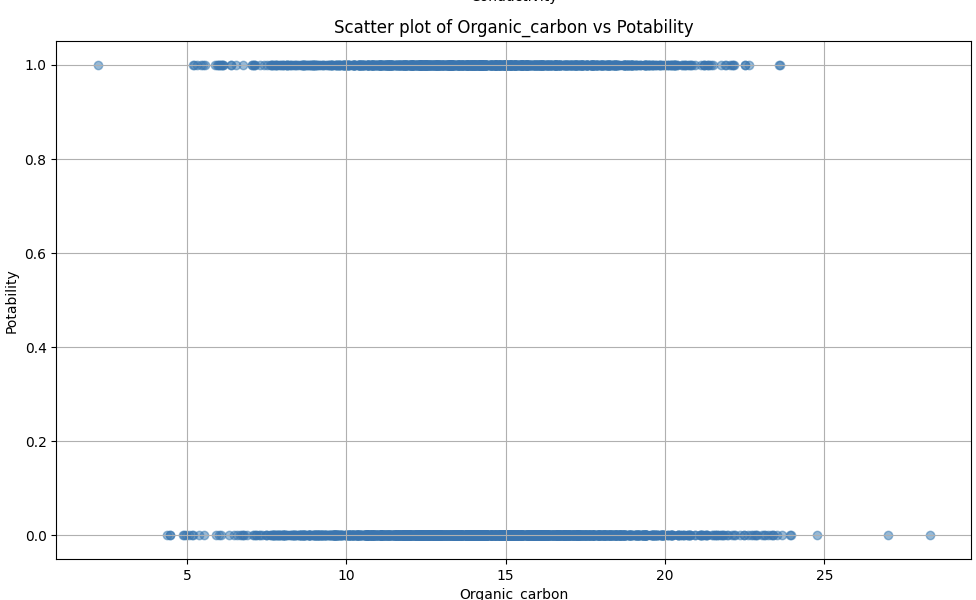
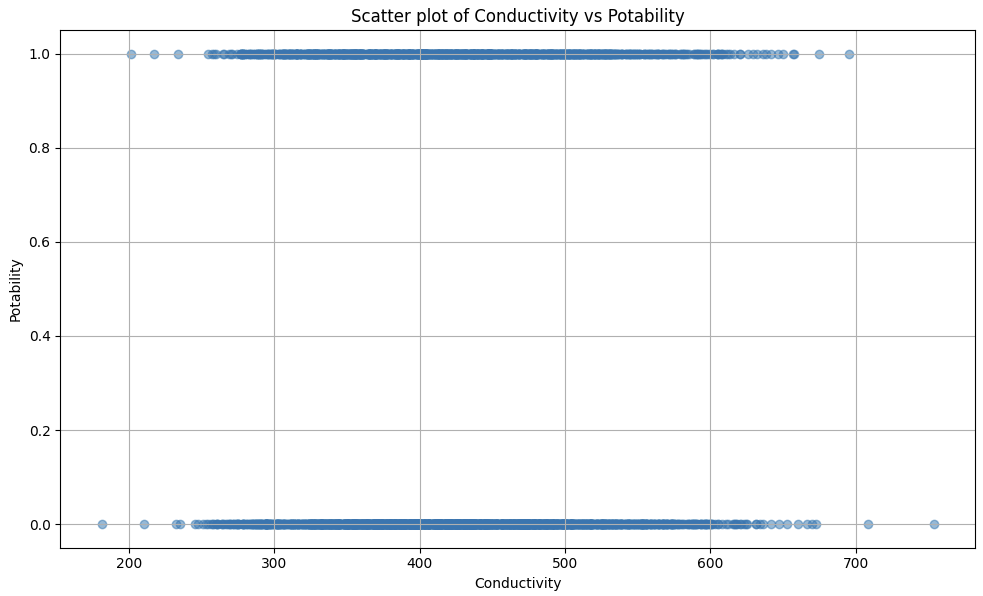
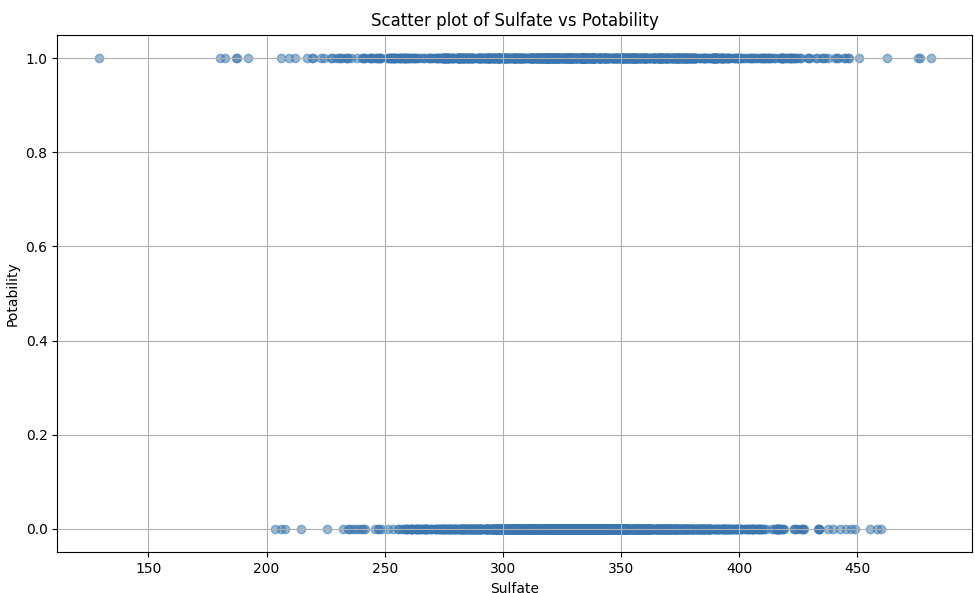
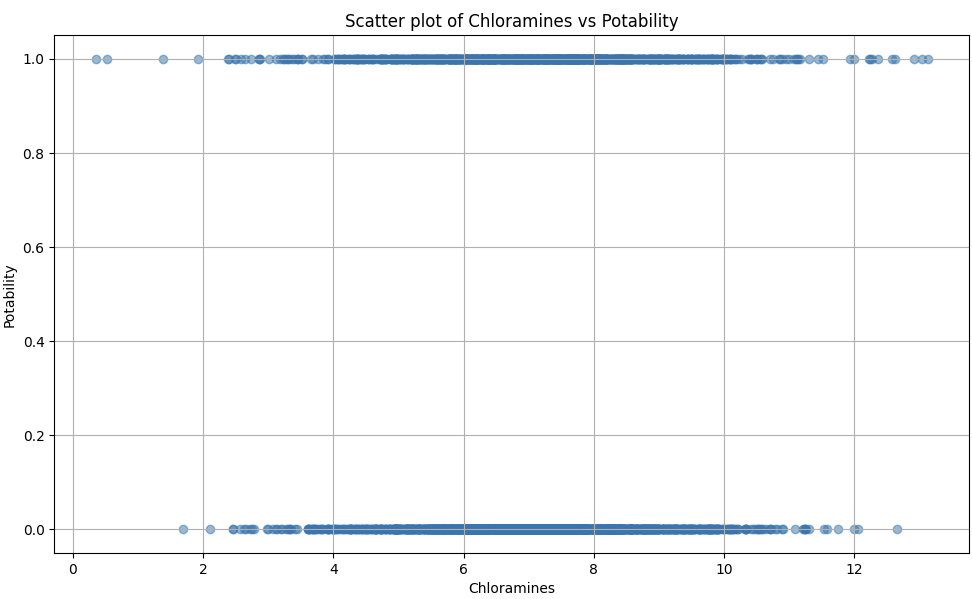
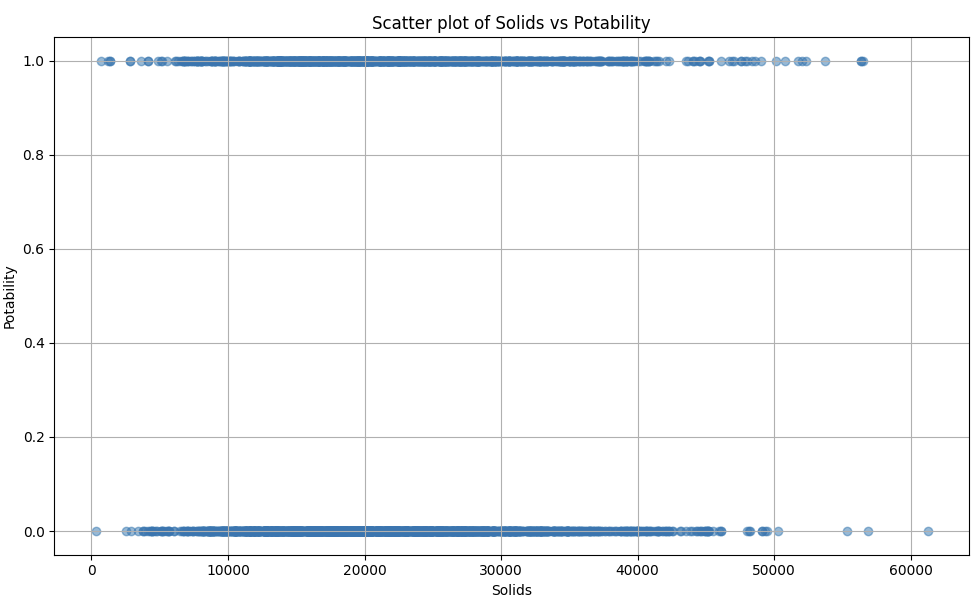
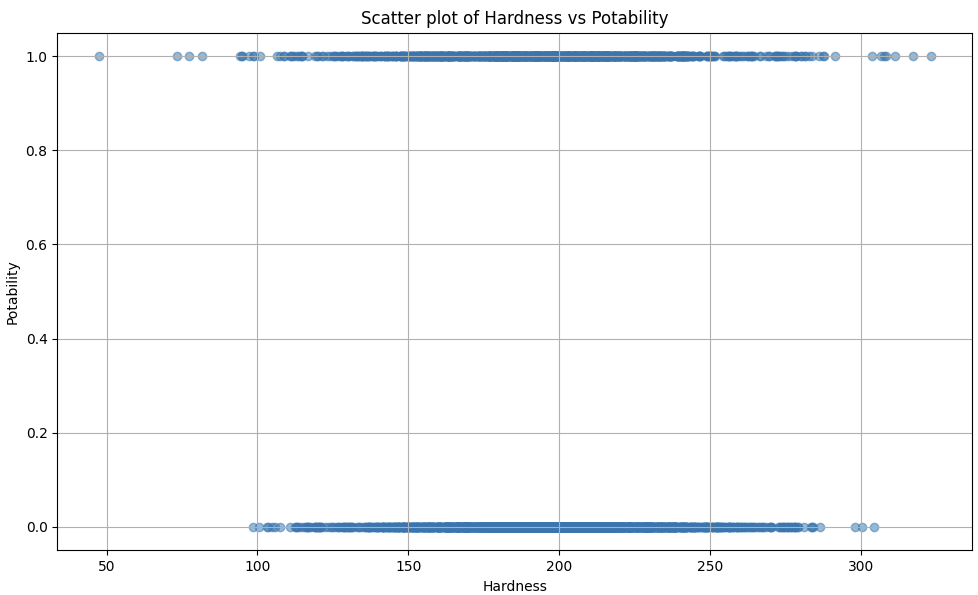
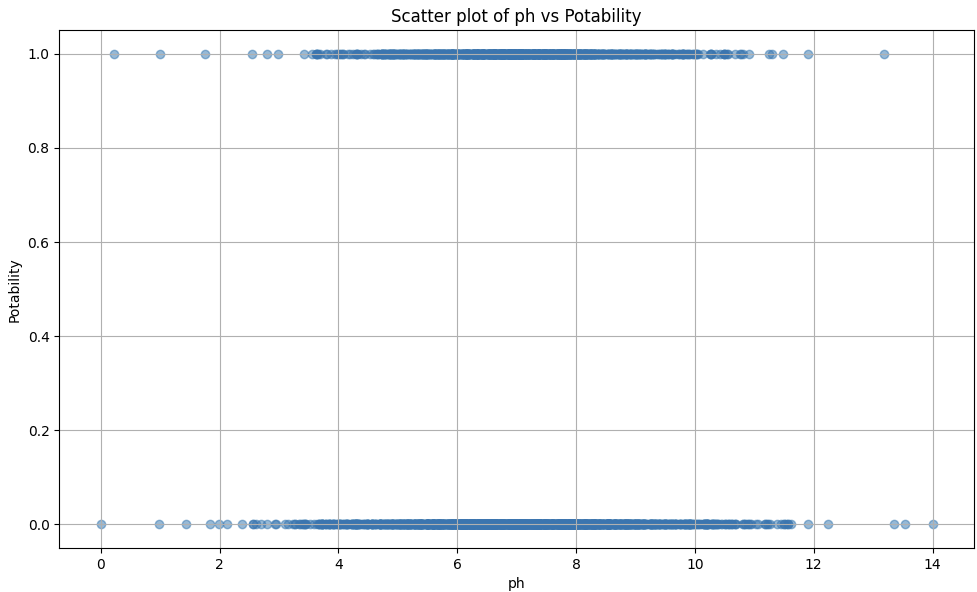
# Turbidity

# Potability

# 

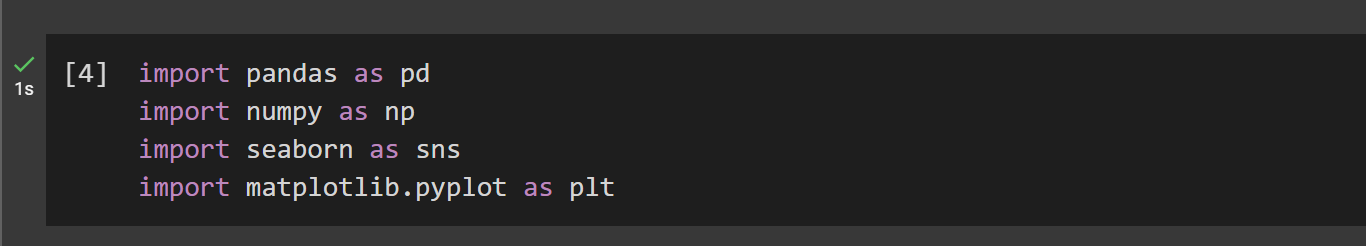
# Dataset:

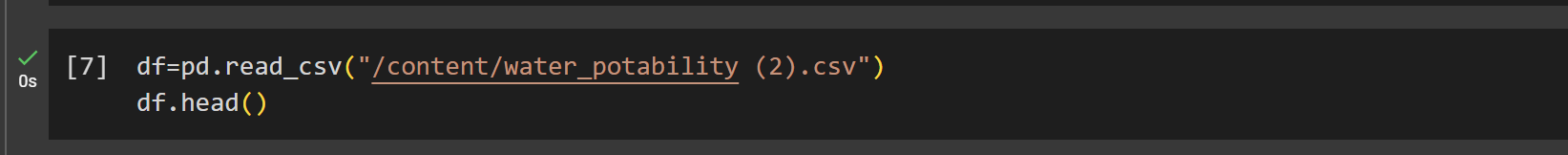
****

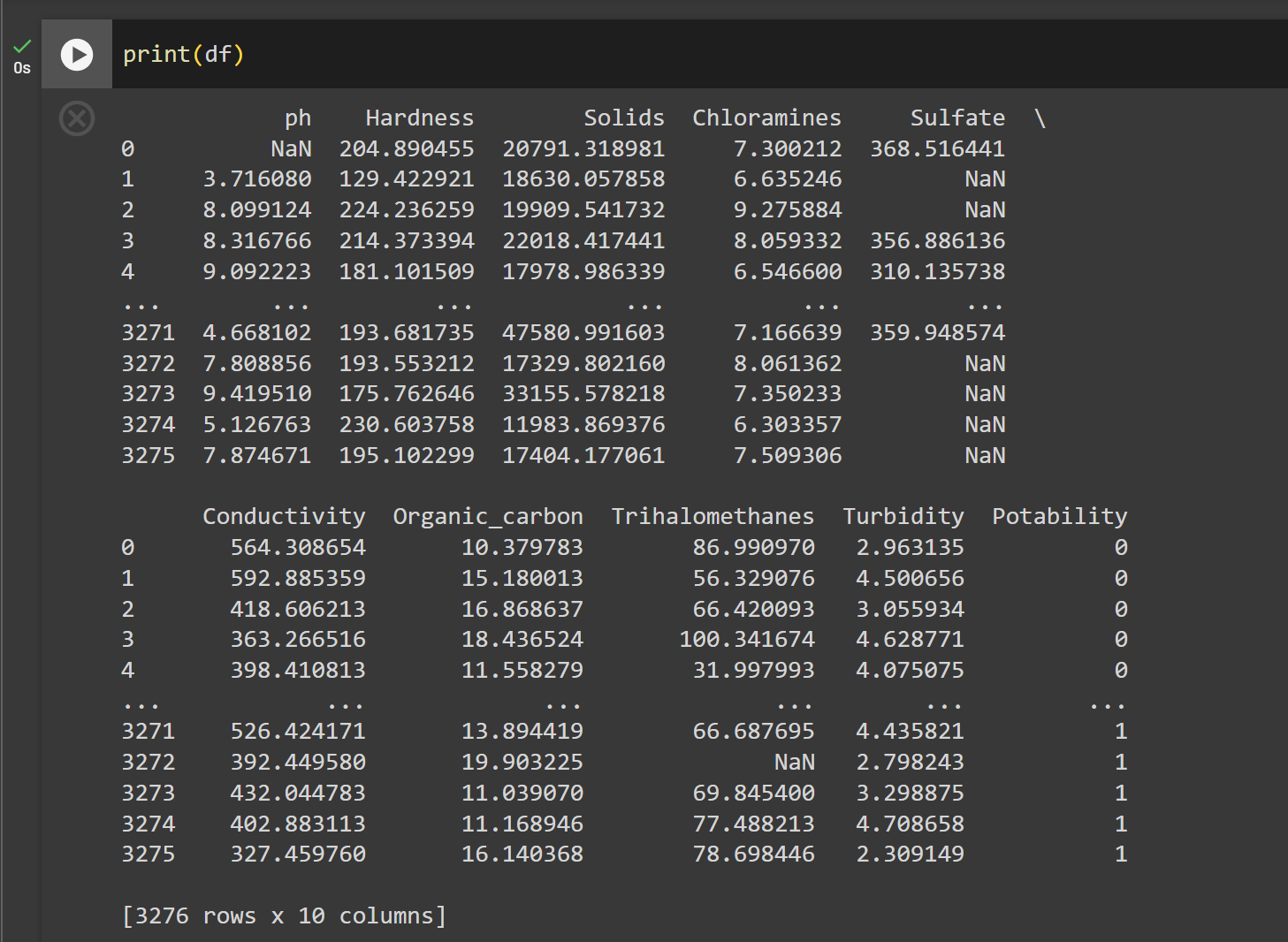
**GRAPHS PLOTTED BETWEEN FEATURE AND TARGET VARIABLES: **

# 5. RESULTS:

**CODE**

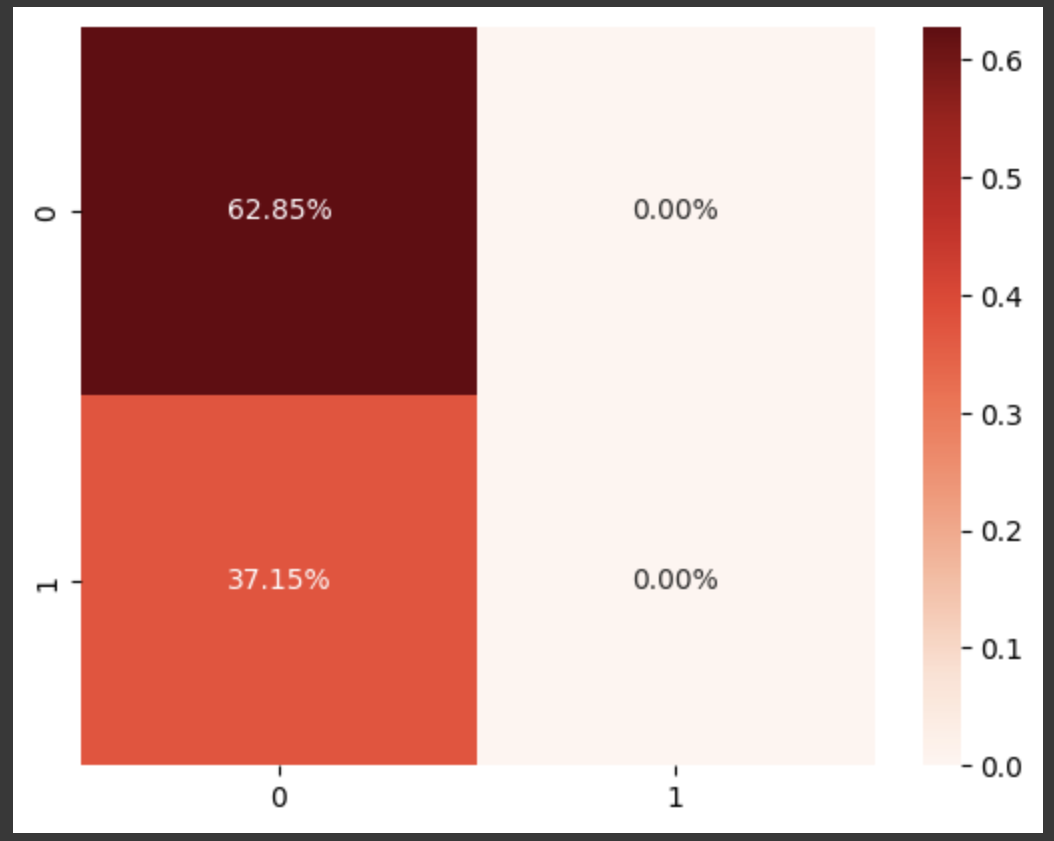
**Dataset: **

****

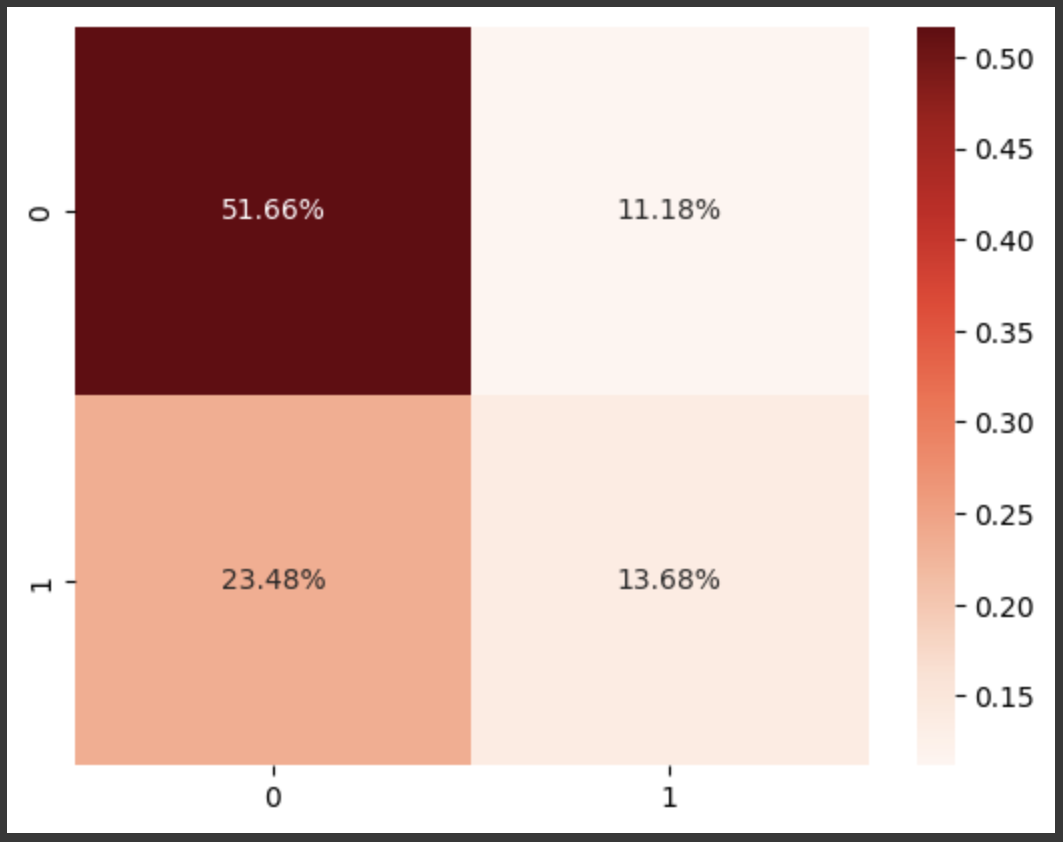
****

**BEFORE APPLYING SMOTE:**

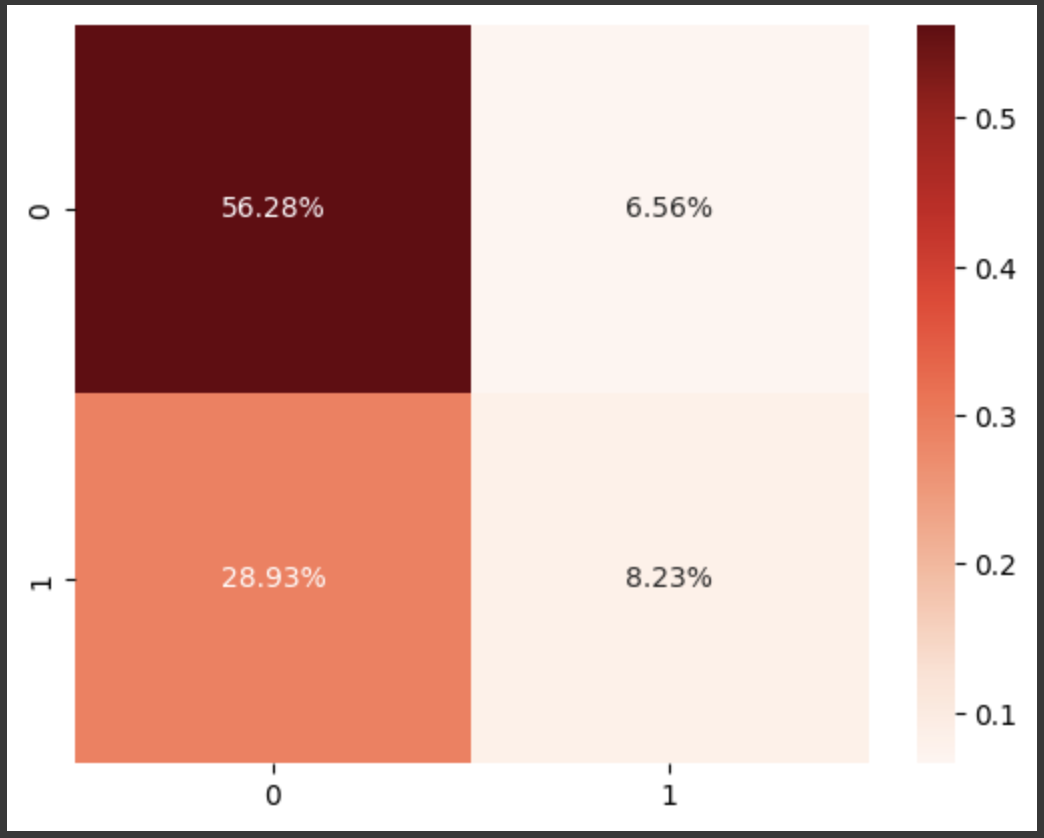
**Logistic Regression:**

****

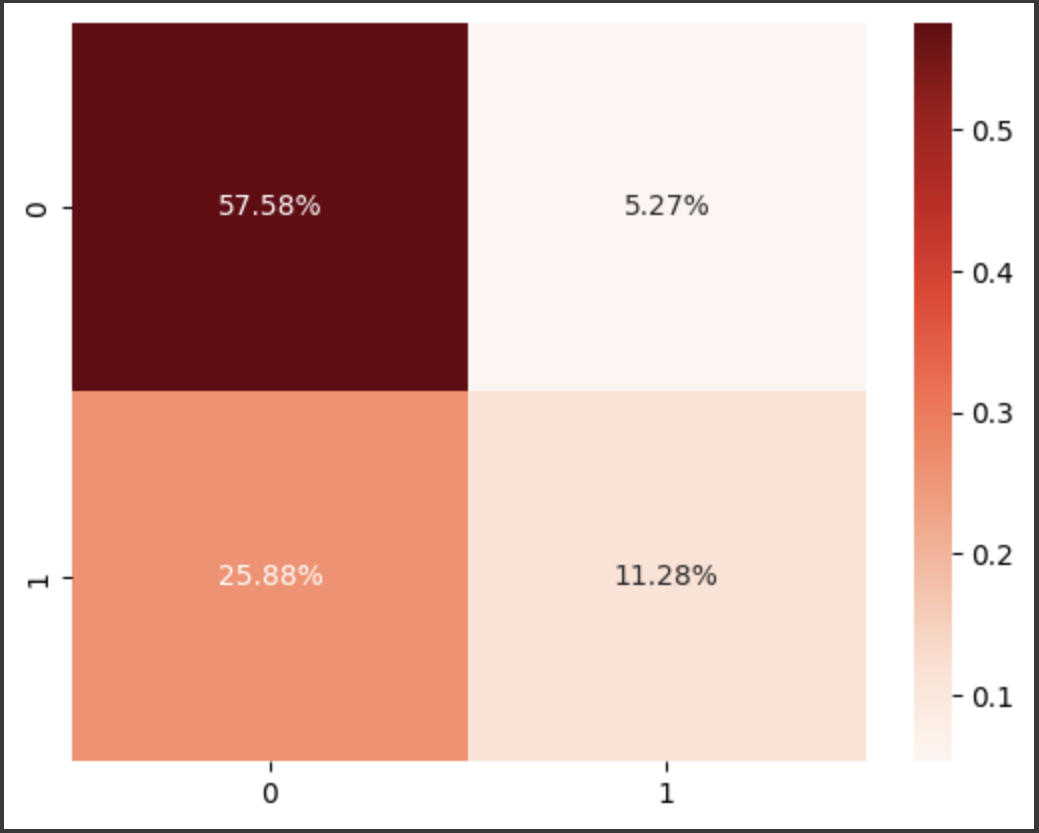
**K Nearest Neighbor:**

****

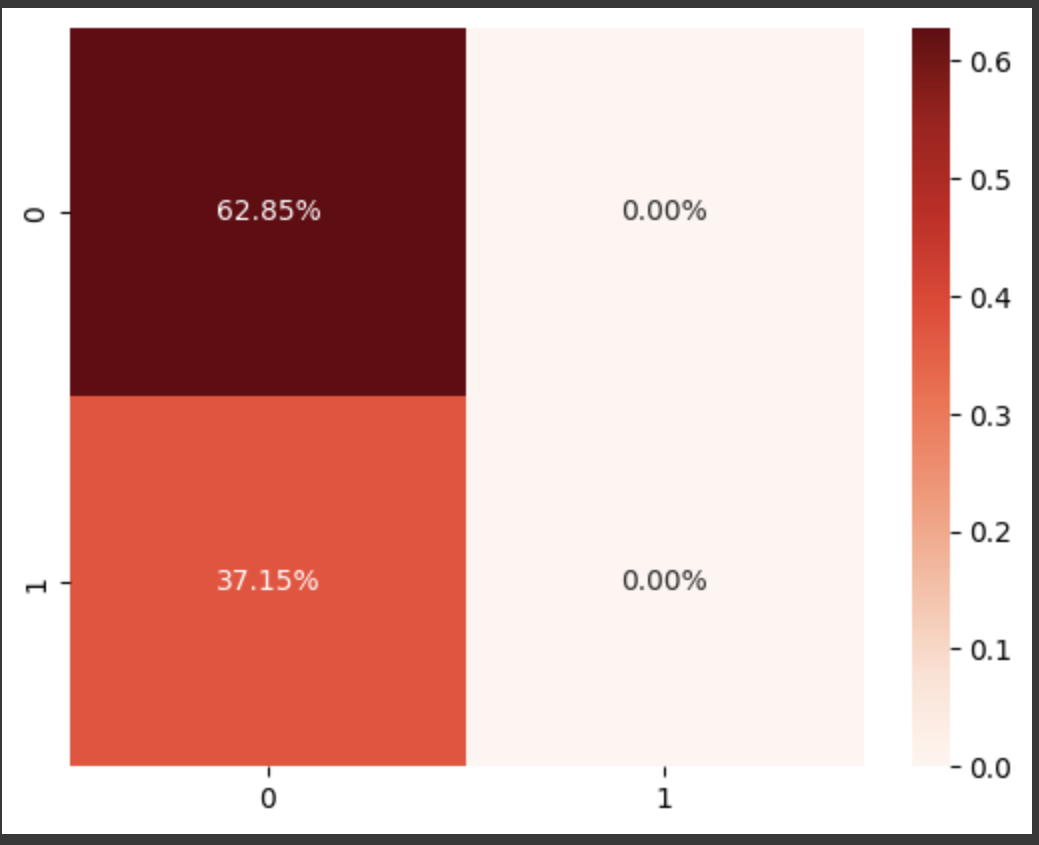
**Decision Tree:**

****

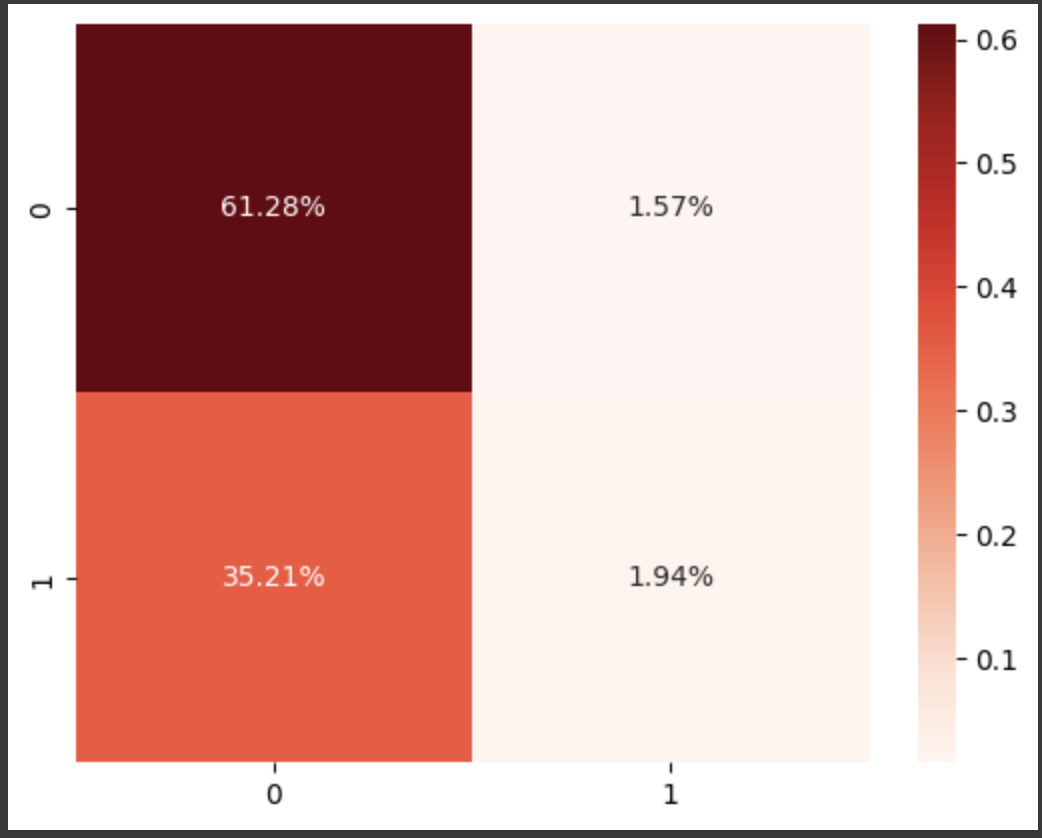
**Support Vector Machine:**

****

**Random Forest:**

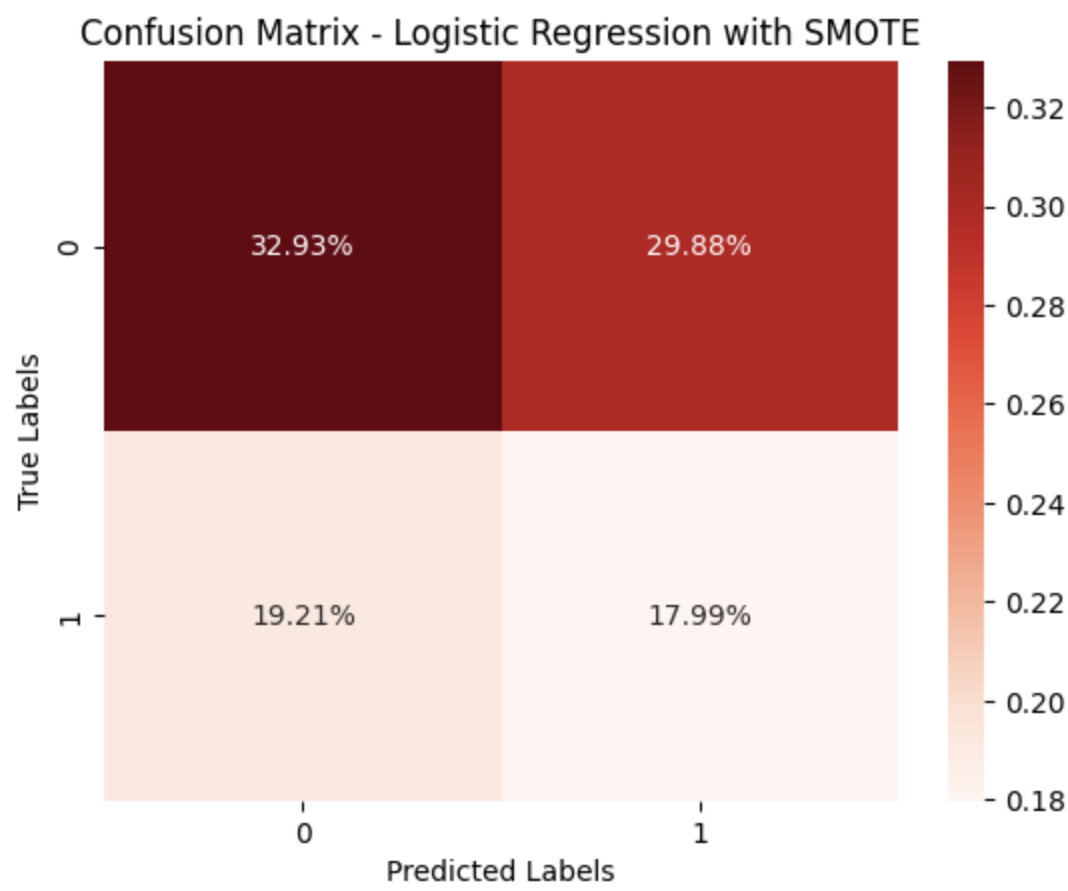
****

**Adaboost:**

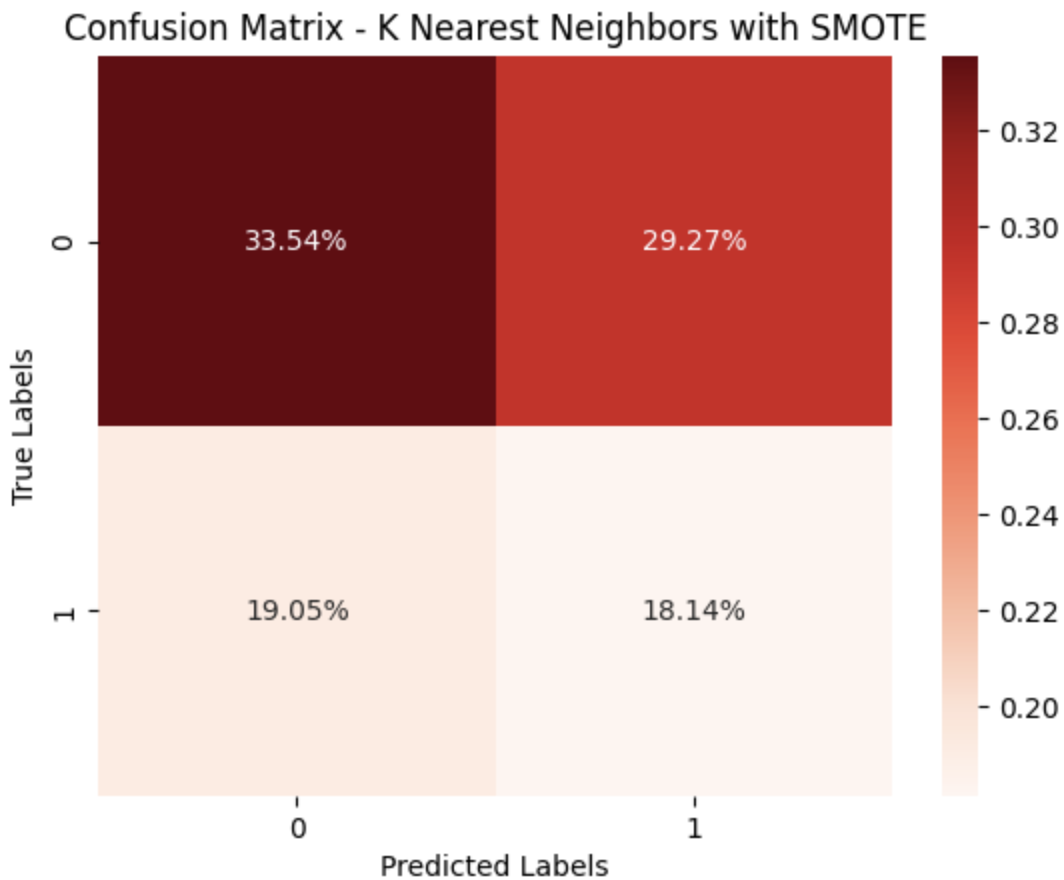
****

**AFTER APPLYING SMOTE:**

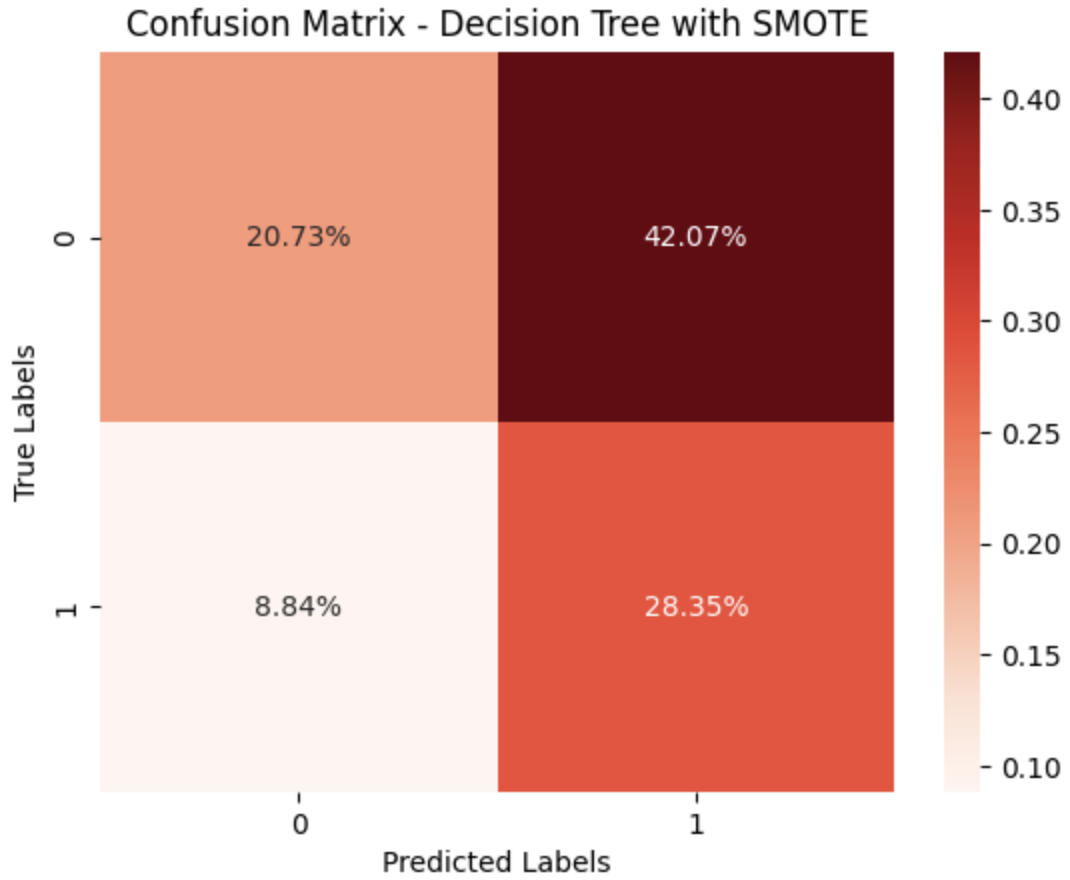
**Logistic Regression:**



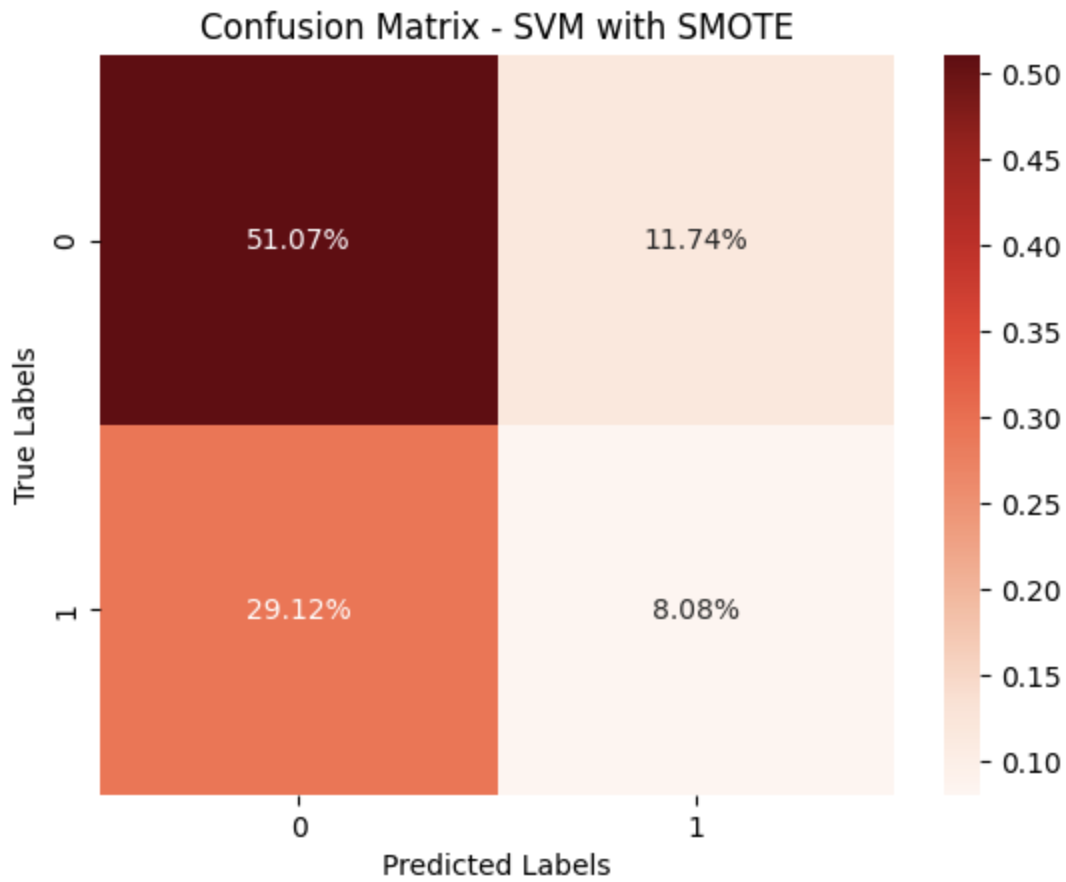
**K Nearest Neighbor:**



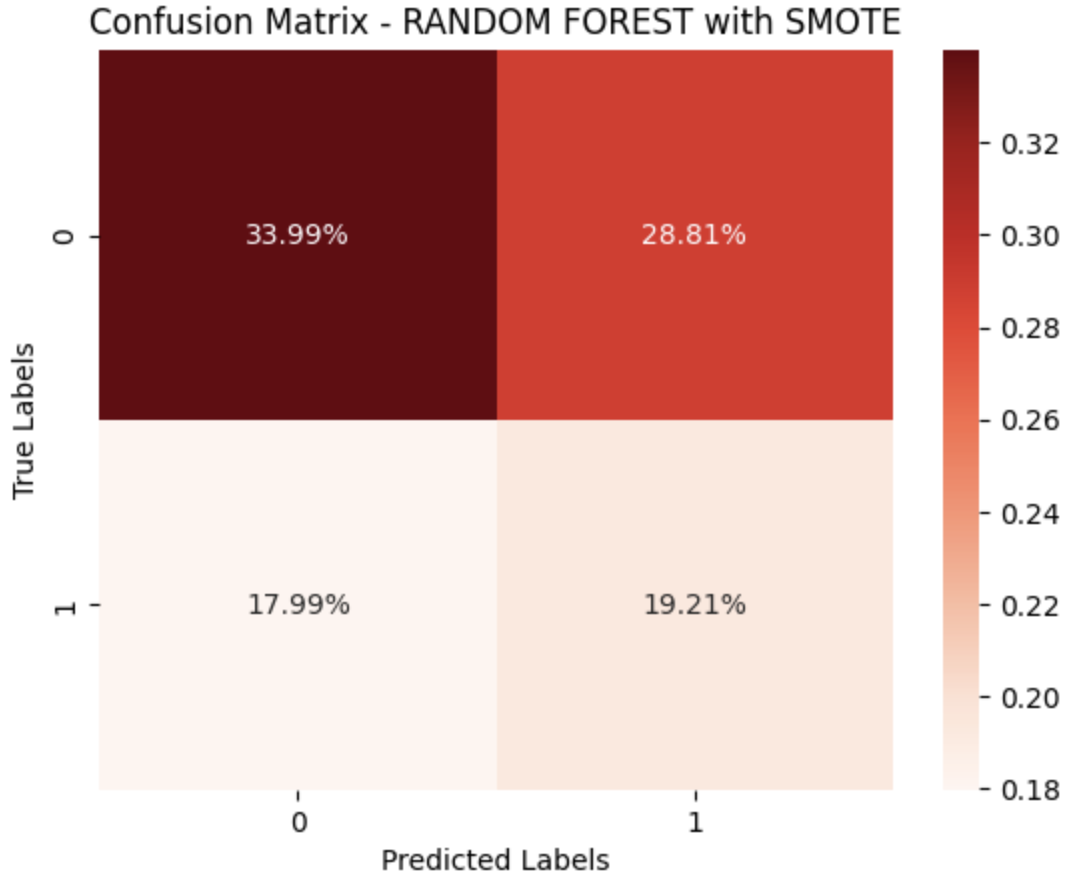
**Decision Tree:**



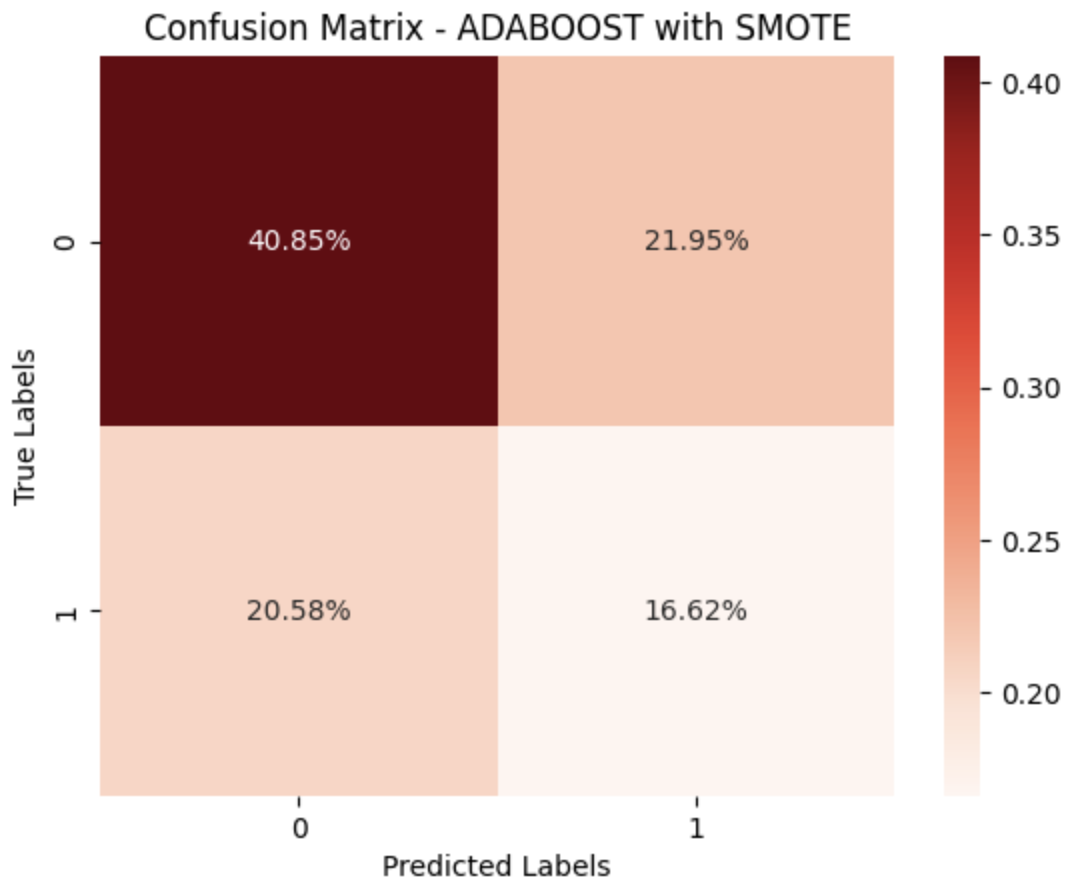
**Support Vector Machine:**



**Random Forest:**



**Adaboost:**



**BEFORE USING SMOTE:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **MACHINE LEARNING MODEL** | **ACCURACY** | **PRECISION** | **RECALL** | **F1-SCORE** |
| 1 | Logistic Regression | 0.6284658040665434 | 0.5364402344995626 | 0.5091463414634146 | 0.5171496692292157 |
| **2** | K-Nearest Neighbour | 0.6534195933456562 | 0.5428170453240949 | 0.5167682926829268 | 0.5245515741601059 |
| **3** | Support vector Machine | 0.6885397412199631 | 0.551634695639147 | 0.5914634146341463 | 0.5540256013713688 |
| **4** | Decision Tree | 0.6451016635859519 | 0.6217805111046067 | 0.6451016635859519 | 0.5954968472594475 |
| **5** | Random Forest | 0.6284658040665434 | 0.3944452706722189 | 0.6280487804878049 | 0.4845619804512653 |
| **6** | Adaboost | 0.6371951219512195 | 0.6476348854397636 | 0.6371951219512195 | 0.5197629243468007 |

# AFTER USING SMOTE:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | MACHINE LEARNING MODEL | **ACCURACY** | **PRECISION** | RECALL | **F1-SCORE** |
| **1** | Logistic Regression | 0.5091463414634146 | 0.37579617834394907 | 0.48360655737704916 | 0.5171496692292157 |
| 2 | K-Nearest Neighbour | 0.5167682926829268 | 0.5428170453240949 | 0.5167682926829268 | 0.5245515741601059 |
| 3 | Support vector Machine | 0.49085365853658536 | 0.551634695639147 | 0.5914634146341463 | 0.5540256013713688 |
| 4 | Decision Tree | 0.5914634146341463 | 0.5900282141795845 | 0.49085365853658536 | 0.4778821107426849 |
| 5 | Random Forest | 0.5320121951219512 | 0.5594986052499822 | 0.5320121951219512 | 0.5396691808040175 |
| 6 | Adaboost | 0.5746951219512195 | 0.5779079897969509 | 0.5746951219512195 | 0.5761976618050364 |

# 6. CONCLUSION:

Before we applied SMOTE, which is a technique to handle imbalanced datasets, we tested several machine learning models. The **Support Vector Machine (SVM) is the best**. It had the highest accuracy, precision, recall, and F1-score. These are all measures of how well the model is performing. In simple terms, SVM was the best at predicting water quality based on our dataset before we used SMOTE.

After we applied SMOTE to our dataset, we tested the same models again. This time, the **Adaboost** model had the highest F1-score. The F1-score is a measure that considers both precision (how many of the positive predictions were actually correct) and recall .So, after using SMOTE, Adaboost was the best at predicting water quality.

In conclusion, if you’re not using SMOTE, the SVM model would be the best choice based on these results. But if you are using SMOTE, then the Adaboost model would be the best choice.

**7. FUTURE SCOPE:**

# Moving forward, this project holds several avenues for expansion and refinement. Future endeavors could involve integrating IoT sensors for real-time monitoring, exploring ensemble learning techniques for improved accuracy, and conducting extensive feature engineering to extract deeper insights. Spatial and temporal analysis, coupled with online learning methods, could enhance the model's adaptability to dynamic environments. This could help analyzing the water quality before using for the people’s needs. This can prevent many health problems that are caused by polluted water.

# 8. REFERENCES:

1. <https://www.kaggle.com/code/imakash3011/water-quality-prediction-7-model/notebook>
2. <https://www.mdpi.com/2073-4441/14/7/1067>
3. <https://www.sciencedirect.com/science/article/pii/S0957582022010>
4. <https://www.mdpi.com/2079-3197/11/2/16>