## Objective

The goal of this project assignment is to build and test classifier models on the data based on water potability, and predict if the water is safe to drink based on different features. Initially, the data set is preprocessed and simple analysis is performed. The data set is readily available on Kaggle, and found in the reference section of the project.

## Preprocessing/Investigation

There are nine continuous features in the data set in addition to the class label. This preprocessing and investigation step is broken into three steps:

1. There is **no correlation** between the features. All correlation coefficients between the features are significantly low, and no features shall be removed from the model.

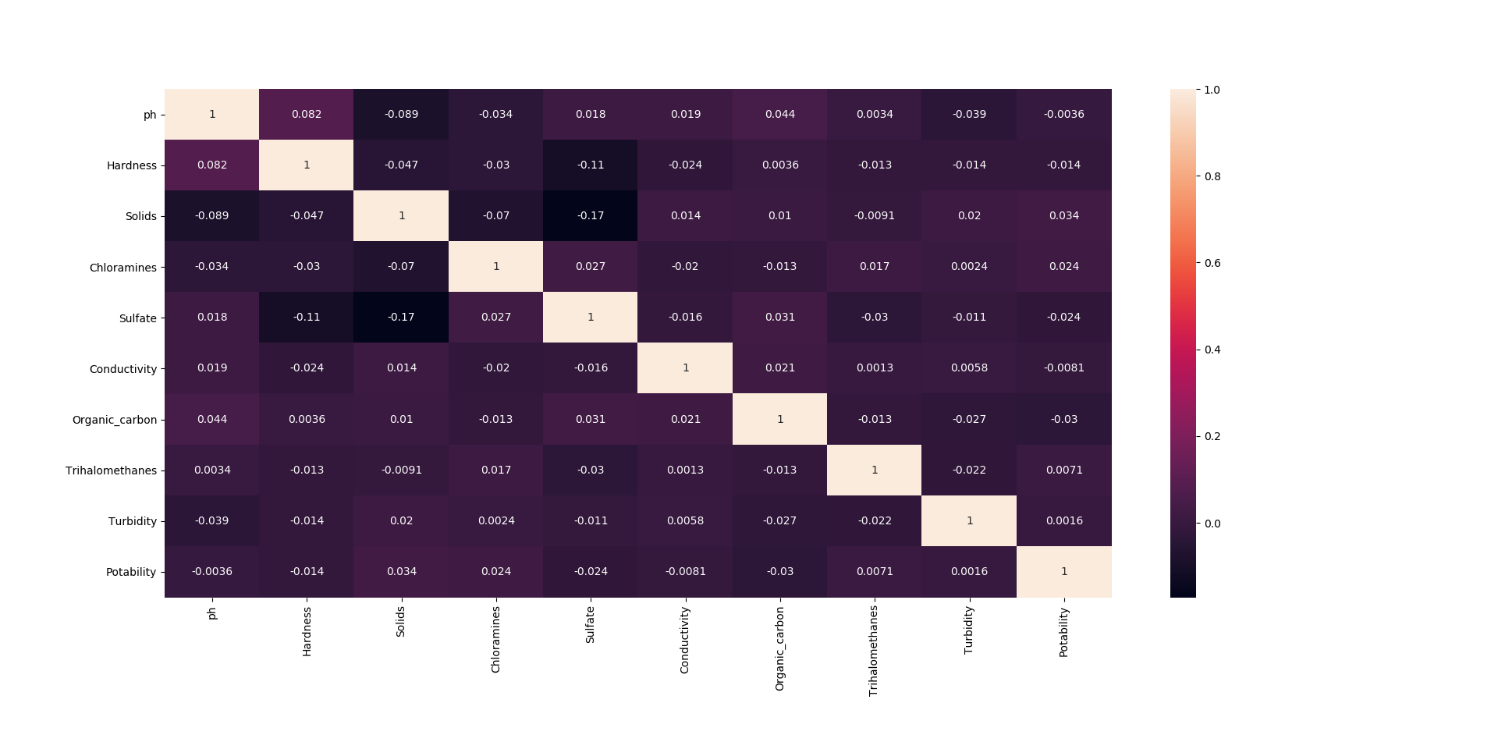


Figure1: Correlation Matrix of all features

1. On visualizing the histograms (as show in the appendix), it is indicated that all features (except class) have a similar to normal distribution.
2. There were three columns with missing data namely ‘ph’, ‘Sulfate’ and ‘Trihalomethanes’. The missing values were filled by the attribute mean for all samples belonging to the same class.

Additional to the above steps, there is class imbalance (slightly skewed distribution where 1998 are 0 and 1278 are 1) observed in the class label (potability). The best measure to study for this data set is precision. Water unsafe for drinking should not be classified as safe (false positive), and it is very important to reduce such cases. F-Score and Recall will also be calculated in this project. The important formulas are displayed below.

= =

=

## Classifiers

Some of the models require standardizing while its optional in others. For simplicity standardization is performed on all the training and testing set except Naïve Bayes. In order to tune the best hyper parameters, GridSearchCV is performed on the training set. In order to find the best hyperparameters, cross validation was also done. Finally, the best estimator used in the training in terms of precision is used to predict labels in the testing set. The following models are listed with their respective best hyper parameters.

* **KNN classifier** was performed for k ranging from 3 to 19 (odd numbers only). The one with the highest precision had k (nearest neighbor) = 19
* **Logistic Regression** with highest precision in training had C = 0.1 and penalty = l2 as its hyperparameters.
* **Naïve Bayes** (No hyperparameters in the same sense as other models)
* **Decision Tree** with highest precision in training had criterion as ‘gini’ and maximum depth = 2.
* **Random Forest** with highest precision in training had maximum depth = 2, minimum sample leaf = 1, minimum sample split = 5 and number of estimators as 500.
* **Support Vector Machines** with highest precision in training had C=100, gamma = 0.001 and kernel as ‘rbf’.

The following table displays the comparison of the six models.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Accuracy | FScore | Precision | Recall | TP | FP | FN | TN | TPR | TNR |
| KNN | 0.6632 | 0.3904 | 0.6385 | 0.2811 | 106 | 60 | 271 | 546 | 0.2811 | 0.9009 |
| Logistic Regression | 0.6215 | 0.0606 | 0.6315 | 0.0318 | 12 | 7 | 365 | 599 | 0.0318 | 0.9884 |
| Naïve Bayes | 0.6368 | 0.3449 | 0.5595 | 0.2493 | 94 | 74 | 283 | 532 | 0.2493 | 0.8778 |
| Decision Tree | 0.7049 | 0.3855 | 0.9578 | 0.2413 | 91 | 4 | 286 | 602 | 0.2413 | 0.9933 |
| Random Forest | 0.6734 | 0.2620 | 0.9827 | 0.1511 | 57 | 1 | 320 | 605 | 0.1511 | 0.9983 |
| SVM | 0.6358 | 0.1050 | 0.9130 | 0.0557 | 21 | 2 | 356 | 604 | 0.0557 | 0.9967 |

Table1: Metrics of all classifiers

Decision Tree and Random Forest classifiers were clearly the pick of the lot. The main idea behind our metric selection is to remove false positive. It can be very dangerous for public if rate of false positive is high. The recall values were also low for the classifiers. Purely based on precision, **Random Forest** should be selected as the best model. In case where F-scores and recall are also considered, Decision Tree stood out the most. KNN did have the highest recall, but the number of false positive surely established that it is not a suitable model for our objective.

Decision Tree had the highest accuracy of 70.49% followed by Random Forest at 67.34% and KNN at 66.32%. Accuracy is definitely not a suitable measure due to the class imbalance and importance of false positives for this project. To overcome the imbalance issue, under-sampling or over-sampling (using SMOTE) is often performed.

Finally, we can conclude the best model selected is Random Forest with highest precision of 0.9827 for this project.

A way of improving the classifier involves feature selection and feature generation. In feature selection, some important features are selected in order to predict the class. The redundant features are removed from the model. While combining features and creating new features from them sometimes increases the performance of the model, and this process is called feature generation.

Note: The gridsearchCV for Random Forest and SVM are commented out in the code, as the process takes much longer to run. The best parameters are selected for them manually, and the metrics are calculated using those hyperparameters.

## Appendix

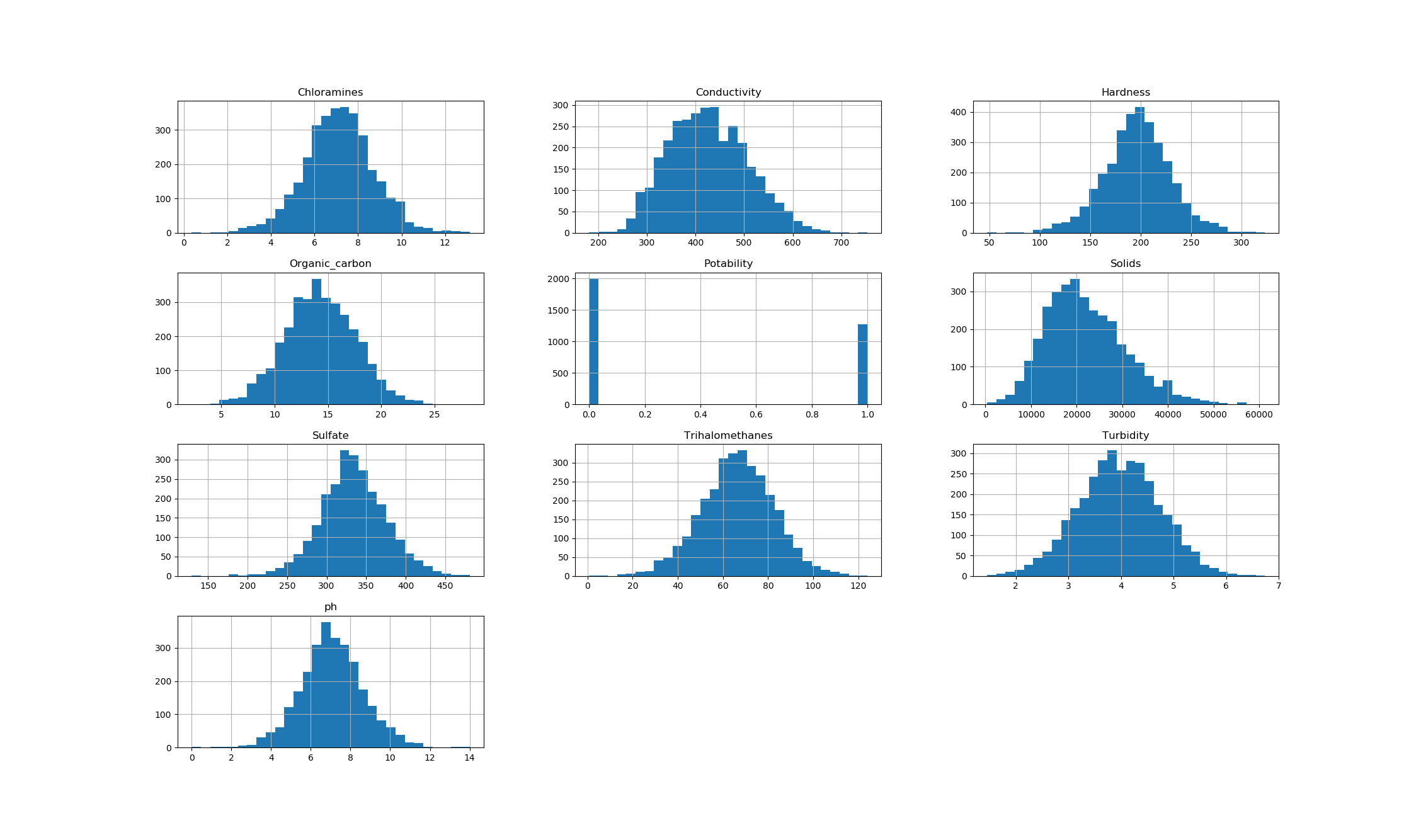


Figure2: Histogram of all features in the data set

## References

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