Table of Contents

[**1 Introduction 1**](#_Toc113296174)

[**1.1 Aims 1**](#_Toc113296175)

[**1.2 Objectives 1**](#_Toc113296176)

[**2 Exploratory Data Analysis (EDA) 2**](#_Toc113296177)

[**3 Missing Value Treatment 6**](#_Toc113296178)

[**4 Feature Selection 7**](#_Toc113296179)

[**4.1 Correlation Heatmap 7**](#_Toc113296180)

[**4.2 Analysis of Variance Test (ANOVA) 8**](#_Toc113296181)

[**4.3 Conclusion 9**](#_Toc113296182)

[**5 Synthetic Minority Over-sampling Technique (SMOTE) 11**](#_Toc113296183)

[**Standardisation (MinMaxScaler) 13**](#_Toc113296184)

[**6.1 Support Vector Machine 17**](#_Toc113296185)

[**6.1.1 Non-Optimised Model 17**](#_Toc113296186)

[**6.1.2 Optimised Models 20**](#_Toc113296187)

[**6.1.3 Summary 23**](#_Toc113296188)

[**References 24**](#_Toc113296189)

# 1 Introduction

Around the world, there exist many sources of water that are contaminated, especially in third-world countries where the sanitation is not good. According to the (World Health Organisation, 2019), approximately 800 million people do not have access to a basic drinking-water facility with a minimum of 2 billion people drinking from faecal-contaminated water sources. Moreover, it is predicted that by 2025, one-half of the world’s population will be forced to live in water-stressed regions (World Health Organisation, 2019). Due to this, potable water, which is another term for water that is safe for drinking has become very important. By conducting a project such as this, water potability can be predicted and governments would be able to immediately identify safe water sources and make them more accessible.

## Aims

The research aims to identify which variables affect the portability of a water source using deep learning models.

## 1.2 Objectives

* To build and utilize various type of deep learning models for predicting water potability
* To identify the features that have an effect on water potability
* To find out the most relevant deep learning model for the business use case

# 2 Exploratory Data Analysis (EDA)

Exploratory data analysis is the very crucial process of exploring the data through preliminary investigations in an attempt to find out any relationships, patterns, anomalies and make a summary of the main characteristics of the dataset (Patil, 2018). The table below describes the different variables present in the dataset.

Table 1: Attribute Description

|  |  |
| --- | --- |
| **Attribute Name** | **Description** |
| **pH** | The pH of water is an important metric in determining its acid–base balance. It can also be used to determine if the water is acidic or alkaline. According to the author, the pH maximum allowable limit has been set at 6.5 to 8.5 by the WHO. This dataset ranges from 6.52 to 6.83, which is within those guidelines (Kadiwal, 2021). |
| **Hardness** | The capacity of water to precipitate soap produced by calcium and magnesium was formerly termed as hardness. However, calcium and magnesium salts have been found to be the main causes of hardness. These salts are dissolved in water as it passes through geologic layers. The amount of time water spends in contact with hardness-producing material affects the amount of hardness in raw water (Kadiwal, 2021). |
| **Solids** | An extensive range of inorganic as well as some organic minerals and salts can be dissolved by water, including calcium, potassium, bicarbonate and chloride. In addition to producing an unwanted flavor, these minerals also dilute the color of water. TDS should not exceed 500 mg/l and the maximum limit should not exceed 1000 mg/l for drinking water (Kadiwal, 2021). |
| **Chloramines** | Among the most common disinfectants used in public water systems are chlorine and chloramine, which commonly develops when ammonia is combined with chlorine to treat water. According to the author, it is considered safe for drinking water to contain up to 4 milligrams of chlorine per liter (mg/L) or 4 parts per million (ppm) (Kadiwal, 2021). |
| **Sulfate** | Minerals, soil, and rocks all include sulphates, which are naturally occurring compounds which can be found in the plants, groundwater and in the environment. Chemical industries are the main users of sulfate. Approximately 2,700 mg of sulfate are found in seawater per liter. In most freshwater supplies, its concentration ranges from 3 to 30 mg/L, although very high concentrations (1000 mg/L) can be found in some regions (Kadiwal, 2021). |
| **Conductivity** | Pure water is a good insulator of electric current. The electrical conductivity of water improves as ions concentration rises. The electrical conductivity of water is usually determined by the amount of dissolved particles in it. Electrical conductivity (EC) is a measurement of a solution's ionic process, which allows it to transmit current. The EC value should not exceed 400 S/cm, according to WHO guidelines (Kadiwal, 2021). |
| **Organic\_carbon** | Naturally decaying organic matter (NOM) and synthetic materials contribute to the total organic carbon (TOC) in source waters. The total quantity of carbon in organic compounds in pure water is measured by the TOC. US EPA recommends a TOC of less than 2 mg/L in drinking water and a TOC of less than 4 mg/L in the source water for treatment (Kadiwal, 2021). |
| **Trihalomethanes** | THMs are compounds that can be found in chlorine-treated water. THM levels in drinking water vary depending on the amount of organic matter in the water, the amount of chlorine needed to treat the water, and the temperature of the water being treated. THM levels in drinking water up to 80 ppm are deemed safe (Kadiwal, 2021). |
| **Turbidity** | The amount of solid stuff in the suspended state determines the turbidity of water. This test is used to estimate the amount of colloidal matter in a waste discharge which measures the light-emitting properties of water. According to the WHO, the recommended value is 5.00 NTU (Kadiwal, 2021). |
| **Potability** | The target variable, which specifies if water is safe for consumption. 1 means that it is safe whereas 0 means that it is not safe for consumption (Kadiwal, 2021). |

A picture containing graphical user interface

Description automatically generated

Figure 1: First 5 records of the dataset

The EDA analysis performed is exploration of the data where the first five records are examined using the “head” function by the pandas library as seen in figure 1.

Table

Description automatically generated with low confidence

Figure 2: Statistics of each variable

The second analysis is where the function “describe” is used to generate descriptive statistics of each variable within the dataset (pandas, 2021). As shown in figure 2, the statistics that are generated are count, mean, standard deviation, minimum, maximum, 1st quarter, 2nd quarter, and 3rd quarter.

Text

Description automatically generated

Figure 3: Summary of the dataset

The function “info” is utilized to illustrate a concise summary about the dataset (pandas, 2021). As described in figure 3 that all the data types are float except the target variable, Portability which is an integer data type.

Text

Description automatically generated

Figure 4: Identify NULL values

The function “isnull” is used to detect missing values on each of the variables within the dataset. Figure 4 shows that out of the ten variables, three of the variables which are pH, Chloramines, and Trihalomethanes has missing value identified. Thus, the treatment of the missing values will be covered in the next chapter.

# 3 Missing Value Treatment

Table

Description automatically generated

Figure 5: Interpolation technique for missing values

In terms of treating missing values, the first technique that is applied is interpolation as shown in figure 5. According to the official documentation, the only suitable method for data frame is the “linear” method and the limit direction parameter is where missing values are filled in the specified direction which is “forward” (pandas, 2021). Thus, after performing interpolation, the variable pH is not fully replaced as there are still some missing values detected.

Text

Description automatically generated with medium confidence

Figure 6: Using Mean imputation to impute the remaining pH values

The technique mean imputation is used to replace the remaining null values in the pH variable. Afterwards, all the missing values has been treated as shown in figure 6.

# 4 Feature Selection

Once the missing value has been treated, feature extraction is conducted to select the most important feature/attribute that affects whether the water is potable or not. This will be done with two techniques: correlation heatmap and Analysis of Variance Test (ANOVA). The statistical measure of a linear relationship between two variables is known as correlation. It's also known as the measure of interdependence between two variables. If there are multiple variables, the goal is to find correlation between all of these variables to understand how each variable correlates with one another to affect the final result (Kumar, 2020). Thus, a correlation heatmap is a graphical depiction of a correlation matrix that shows how different variables are related. According to analyticsvidhya.com (2020), ANOVA can be regarded of as a generalization of the t-tests for more than two groups and each independent t-test is used to compare the means of a condition between both groups. However, ANOVA only tests if there is a difference in the mean somewhere in the model but does not tell the difference of it (analyticsvidhya.com, 2020).

## 4.1 Correlation Heatmap

Timeline

Description automatically generated

Figure 7: Correlation Heatmap

The figure above shows the construction of the correlation heatmap for the treated dataset. The “potability” attribute is dropped before correlating and building the heatmap. This is because, by removing the target variable, the heatmap shows the correlation between the attributes. Thus, if two attribute has high correlation between one another, both features are considered to have similar attribute that affects the target variable and one of the features can be dropped as both attributes provide the similar effect. Based on the observation of the correlation heatmap in Figure 7, there are no feature that highly correlates with one another hence, there are no feature that will be dropped as all features are independent of one another.

## 4.2 Analysis of Variance Test (ANOVA)

A screenshot of a computer

Description automatically generated with medium confidence

Figure 8:Feature Selection techniques

*Source: https://towardsdatascience.com/learn-how-to-do-feature-selection-the-right-way-61bca8557bef*

The ANOVA test is chosen due to figure above and the dataset used. The data set used contains input variable of numerical value and the target variable of output variable. Hence, according to the figure ANOVA is selected due to the dataset having a linear relationship.

Graphical user interface, text, application, email

Description automatically generated

Figure 9:ANOVA Feature Extraction

In the ANOVA test, firstly, the dataset is split into two: one containing all the input feature and one the target feature. Then, the datasets will be fitted with both datasets and the “SelectKBest” method is used to select the feature according to the k-highest score. The score is calculated with the “f\_classif” score function whereby the score function will return F-value between label/feature for classification tasks. Next, the score of each feature is ranked in descending order according to the score importance.

## 4.3 Conclusion

To conclude, the ANOVA method will be chosen to determine the feature selection due to that according to the correlation heatmap, no feature will be dropped as each feature is unique with regards of affecting the target variable thus, not providing a clear view of the feature importance. However, ANOVA provided the importance of each feature against the target variable and such each feature is ranked and chosen. The final feature that will be chosen is the “Solids”, “Organic\_Carbon”, “Chloramines”, “Sulfate”, “Hardness”, “Trihalomethanes”, “Conductivity”, and “pH”. Despite the low importance of the “pH” according to the ANOVA output, it is still selected due to the human health as despite all other feature indicates safe potability of the water, it the pH value of the water is low, it will still cause major issue to the human health as it will be to acidic/alkaline for consumption.

# 5 Synthetic Minority Over-sampling Technique (SMOTE)

Initially, the distribution of non-portable and portable records is unbalanced. This can be verified and visualized by plotting a simple boxplot graph as shown below:

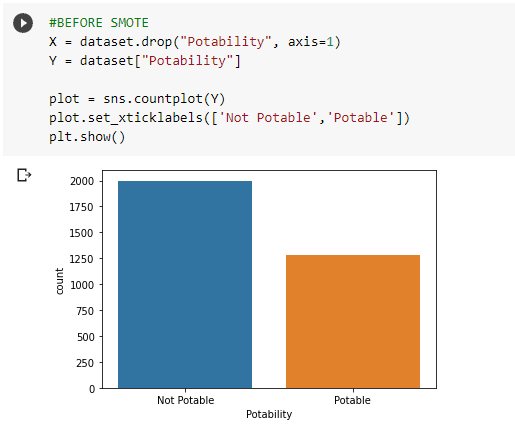


Figure 10: Target variable (Portability) count visualization

The imbalance of data can be detrimental to the accuracy of machine and deep learning models that are to be developed. This is because the data is skewed, and the models are only as good as the data that is put inside of them. Hence, assuming that no corrective action is taken, all the trained models will not have enough sample information regarding whether the water is portable, which consequently means that the accuracy pertaining to such detections will be reduced. This is undesirable as this will result in the accuracy of the networks will be low (Wijaya, 2020). Hence, the synthetic minority over-sample technique is applied to generate pseudo data such that the number of records for portability is equal to that of not portable.



Figure 11:SMOTE Balancing

The Smote implementation comes from the imblearn library and is executed via the smote.fit\_resample function (Imbalanced Learn, 2021). The output of the function is an array which needs to be converted back into a data frame for compatibility of further processing. Once this is complete, a new box plot is generated to validate that the two Y classes have equal count.

Now that smote has been applied, the dataset is now balanced which means that the model will have an equal number of samples to learn from, thus reducing the probability of the models being biased or skewed, which in turn, increases their accuracy.

# Standardisation (MinMaxScaler)

Fundamentally, machine learning algorithms have a better performance when numerical input variables undergo a scaling (Brownlee, 2020). As a best practice in data pre-processing step, when working with machine learning algorithms, there are several methods that can be applied to scale the numeric data. According to the illustration below, MinMaxScaler is used to scale the numeric features in our dataset.

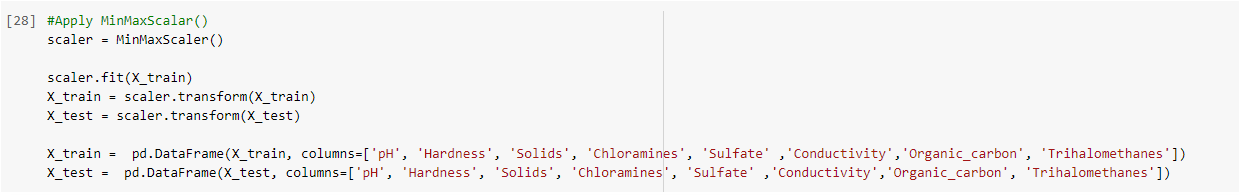


Figure 12:MinMaxScaler

MinMax technique is applied using the sklearn python library whereby it transforms the numeric feature by scaling it to a set range where the default range is 0 to 1.

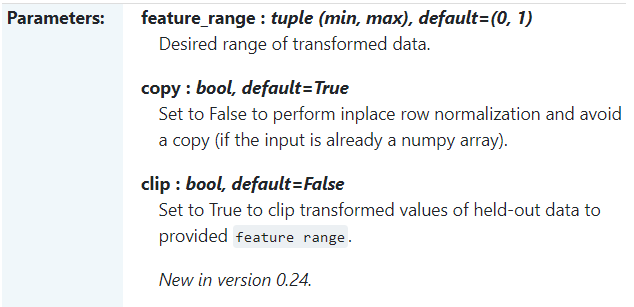


Figure 13: Parameters of MinMaxScaler

How this technique works is by obtaining the standard deviation of respective numeric feature values as well as its minimum and maximum value (sklearn, 2021). These numbers are used to transform and scale the data using the formula given below.

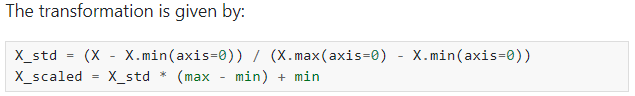


Figure 14:MinMaxScaler Formula

Before MinMaxScaler has been applied, the numeric features look according to the illustration below.



Figure 15:Before MinMaxScaler transformation

After MinMaxScaler has been applied, the features have been scaled to a range between 0 and 1 as shown below.

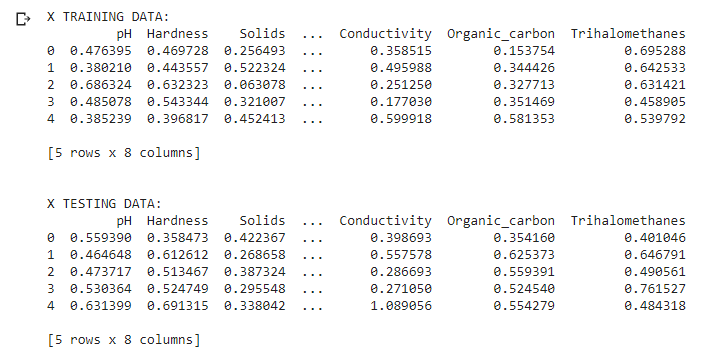


Figure 16: Results of MinMaxScaler transformation

Distribution density analysis has been done for all the selected numeric features that is going to be utilized to train the model. The illustration below shows the distribution for the numeric features before MinMaxScaler is applied.

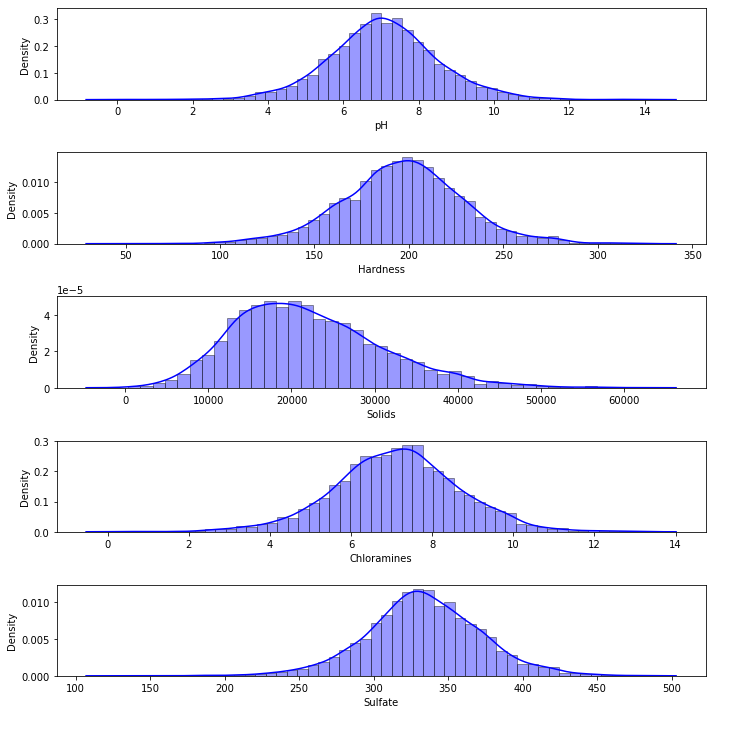


Figure 17: Distribution Density after MinMaxScaler transformation

After MinMaxScaler is applied, the features more closely resemble the bell curve, hence are more normally distributed. This is shown in the illustration below.

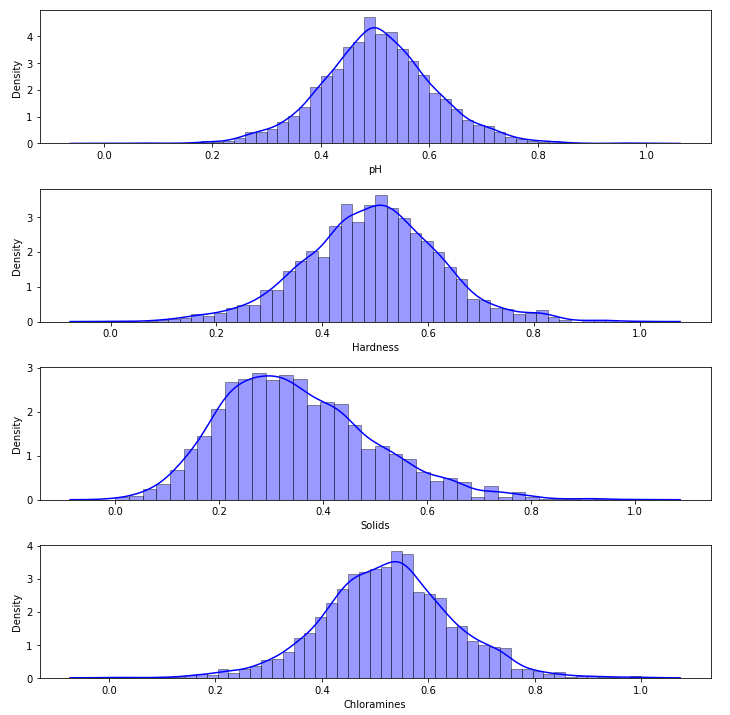


Figure 18: Distribution Density after MinMaxScaler transformation

# 6.1 Support Vector Machine

Support Vector Machine is selected due to the target variable being binary, for which SVM is suitable in predicting.

## 6.1.1 Non-Optimised Model

Table

Description automatically generated

Figure 44: Non-optimised SVM model classification report

Initially, a non-optimised SVM model is built to act as a control for the rest of the models that will be built and evaluated. This is done using the “svm.SVC” method. Firstly, the model is fit with the training sets and then it is used to predict the values of the “X\_test” variable. The model is then evaluated based on its ability to predict “y\_test” which is the test values. As seen in the figure above, the accuracy of the model came to 65.25% with the precision coming to 65.27%, the recall to 65.25% and the F1-score to 65.24%.

Graphical user interface, text, application

Description automatically generated

Chart, line chart

Description automatically generated

Figure : SVM AUC

In order to evaluate the model further, two visualization processes will be utilized. The first is a Receiver Operating Characteristics (ROC) curve, which is a plot displaying the performance of a model in all classification levels (Google Developers, 2021). This is done by plotting the true positive rate against the false positive rate, which is automatically calculated. The Area Under Curve (AUC) positions a random positive sample higher than a random negative sample (Google Developers, 2021). It is therefore used to measure how good the model is at differentiating between classes, with a higher AUC meaning the model can predict and differentiate between 0 classes and 1 classes better (Narkhede, 2018). As can be seen above, the AUC of the basic SVM model is calculated to be 0.65.

Text

Description automatically generated

Diagram

Description automatically generated with low confidence

Figure : Basic SVM model confusion matrix

The second visualization evaluation used is a confusion matrix, which is used to compare the actual target values against the ones that are obtained from the prediction performed by the model which give a rounded view into the performance of the model as well what types of errors are being made (Bhandari, 2020). The confusion matrix for the non-optimised SVM model can be seen in the figure above, where it outputs the true positive, true negative, false positive, and false negative values.

## 6.1.2 Optimised Models

**6.1.2.1 GridSearchCV**

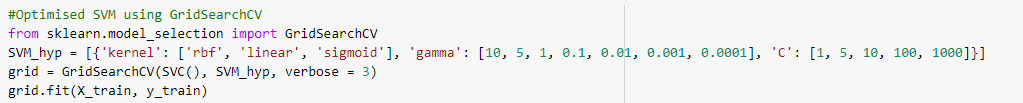


Figure 47: Optimised SVM using GridSearchCV

The first optimized model utilized for SVM used GridSearchCV, which makes use of cross-validation to find the best combination of hyper-parameter values that are fed into the model’s dictionary (Okamura, 2020). As can be seen from the figure above, the hyperparameters used are kernel, gamma and C.

Kernel refers to a group of mathematical functions which are used to transform data into a suitable form (Navlani, 2019). Some of the most commonly used kernels include Radial Basis Function (RBF), linear, and sigmoid. Therefore, these are all used in the dictionary as seen above. The C parameter acts as a regularization parameter, whereby it compromises maximizing the margin of the decision function against correct classifications (Scikit-learn, 2021). This means that it is used to determine how much of a mistake is acceptable. A lower C value encourages a bigger margin and, as a result, a simpler decision function, albeit at the expense of training accuracy. On the other hand, a larger C value means the decision function is better at accurately classifying training samples (Scikit-learn, 2021). Lastly, gamma refers to the impact that a single training sample reaches, with lower values meaning high impact and a tightly-fit model and higher values meaning lower impact and a loosely-fit model.

The best hyperparameter values are then printed and used in the optimized SVM model.

Table

Description automatically generated

Figure 48: Optimised SVM model classification report results

As can be seen above, the best values are obtained using the “best\_params\_” function. The classification report above shows the results obtained from the optimized SVM model using GridSearchCV. The accuracy, precision, recall and F1-score of the model all came to 65.25%.

Chart, diagram

Description automatically generatedChart, line chart

Description automatically generated

Figure 49: GridSearchCV SVM model visualisation evaluation

The AUC of the ROC curve for the optimised SVM model using GridSearchCV computes to 0.65, which is the same as the non-optimised SVM model shown earlier.

**6.1.2.2 RandomizedSearchCV**

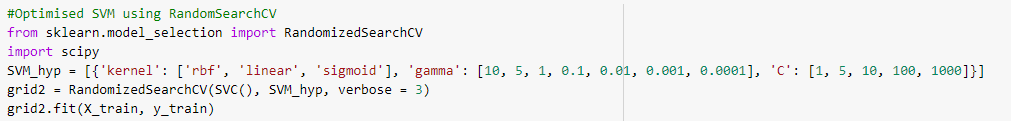


Figure 50: Optimised SVM using RandomizedSearchCV

The RandomizedSearchCV attempts to find the best hyperparameter values by matching them up in random combinations which is slightly similar grid search in GridSearchCV (Ayuya, 2021). This is done using the “RandomizedSearchCV” function. The hyperparameters used are the same ones used for the GridSearchCV optimised model.

Text, table

Description automatically generated

Figure 51: RandomizedSearchCV SVM model classification report

The figure above shows the classification report obtained from using the RandomizedSearchCV SVM model. As can be seen above, the accuracy, precision and F1-score of the model all came to 66.25% with the recall coming to

Chart

Description automatically generatedChart, line chart

Description automatically generated

Figure : RandomizedSearchCV SVM model visualisation evaluation

The AUC of the ROC curve for the optimised SVM model using GridSearchCV computes to 0.66, which is slightly better than the non-optimised SVM model shown earlier.

## 6.1.3 Summary

Table 4: Evaluation Summary

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Basic SVM** | **GridSearchCV** | **RandomizedSearchCV** |
| **Accuracy** | 65.25% | 65.25% | 66.25% |
| **Precision** | 65.27% | 65.25% | 66.26% |
| **Recall** | 65.25% | 65.25% | 66.25% |
| **F1-score** | 65.24% | 65.25% | 66.25% |
| **AUC** | 0.65 | 0.65 | 0.66 |

As seen in the table above, a summary of the evaluation metrics for all of the models built can be seen. As can be seen, only the RandomizedSearchCV optimised model returned better results than the non-optimised, with the GridSearchCV optimised model returning almost similar values. The best results were obtained through the use of RandomizedSearchCV. Better results may be obtained by training the model more and using more data.

# References

analyticsvidhya.com. (2020, June 8). *Introduction to ANOVA for Statistics and Data Science (with COVID-19 Case Study using Python)*. Retrieved from analyticsvidhya.com: https://www.analyticsvidhya.com/blog/2020/06/introduction-anova-statistics-data-science-covid-python/

Ayuya, C. (2021, March 30). *Using Random Search to Optimize Hyperparameters*. Retrieved from Section: https://www.section.io/engineering-education/random-search-hyperparameters/

Bhandari, A. (2020, April 17). *Everything you Should Know about Confusion Matrix for Machine Learning*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2020/04/confusion-matrix-machine-learning/

Brownlee, J. (2020, June 10). *How to Use StandardScaler and MinMaxScaler Transforms in Python*. Retrieved from Machine Learning Mastery: https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/

Capgemni. (2017, October 4). *Quality Data, a must have for AI*. Retrieved from With the advent of Artificial Intelligence (AI) we are able to analyse data in more depth than ever before.: https://www.capgemini.com/2017/10/quality-data-a-must-have-for-ai/

Google Developers. (2021). *Classification: ROC Curve and AUC*. Retrieved from Machine Learning Crash Course: https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc

Imbalanced Learn. (2021). *SMOTE*. Retrieved from Imbalanced Learn: https://imbalanced-learn.org/stable/references/generated/imblearn.over\_sampling.SMOTE.html

Kadiwal, A. (2021, April 25). *Water Quality*. Retrieved from Kaggle: https://www.kaggle.com/adityakadiwal/water-potability

Koehrsen, W. (2018, January 10). *Hyperparameter Tuning the Random Forest in Python*. Retrieved from Towards Data Science: https://towardsdatascience.com/hyperparameter-tuning-the-random-forest-in-python-using-scikit-learn-28d2aa77dd74

Kumar, A. (2020, September 29). *Correlation Concepts, Matrix & Heatmap using Seaborn*. Retrieved from vitalflux.com: https://vitalflux.com/correlation-heatmap-with-seaborn-pandas/

Maladkar, K. (2018, June 14). *Why Is Random Search Better Than Grid Search For Machine Learning*. Retrieved from analyticsindiamag.com: https://analyticsindiamag.com/why-is-random-search-better-than-grid-search-for-machine-learning/

Mujtaba, H. (2020, September 29). *Hyperparameter Tuning with GridSearchCV*. Retrieved from www.mygreatlearning.com: https://www.mygreatlearning.com/blog/gridsearchcv/

Narkhede, S. (2018, June 27). *Understanding AUC - ROC Curve*. Retrieved from Towards Data Science: https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Navlani, A. (2019, December 28). *Support Vector Machines with Scikit-learn*. Retrieved from Datacamp: https://www.datacamp.com/community/tutorials/svm-classification-scikit-learn-python

Okamura, S. (2020, December 29). *GridSearchCV for Beginners*. Retrieved from Towards Data Science: https://towardsdatascience.com/gridsearchcv-for-beginners-db48a90114ee

pandas. (2021). *pandas.DataFrame.describe*. Retrieved from pandas: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.describe.html

pandas. (2021). *pandas.DataFrame.info*. Retrieved from pandas: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.info.html

pandas. (2021). *pandas.DataFrame.interpolate*. Retrieved from pandas: https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.interpolate.html

Patil, P. (2018, March 24). *What is Exploratory Data Analysis?* Retrieved from Towards Data Science: https://towardsdatascience.com/exploratory-data-analysis-8fc1cb20fd15

Pedregosa, F. a. (2011). Scikit-learn: Machine Learning in {P}ython. *Journal of Machine Learning Research*, 2825--2830. Retrieved from sklearn.metrics.precision\_recall\_fscore\_support: https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision\_recall\_fscore\_support.html

Ranjan, C. (2019, May 8). *Understanding Dropout with the Simplified Math behind it*. Retrieved from Towards Data Science: https://towardsdatascience.com/simplified-math-behind-dropout-in-deep-learning-6d50f3f47275

Scikit Learn. (2021). *sklearn.ensemble.RandomForestClassifier*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Scikit Learn. (2021). *sklearn.ensemble.RandomForestClassifier*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Scikit Learn. (2021). *sklearn.model\_selection.RandomizedSearchCV*. Retrieved from Scikit Learn: https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.RandomizedSearchCV.html

Scikit-learn. (2021). *RBF SVM parameters*. Retrieved from Scikit-learn: https://scikit-learn.org/stable/auto\_examples/svm/plot\_rbf\_parameters.html

sklearn. (2021). *sklearn.preprocessing.MinMaxScaler*. Retrieved from sklearn: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

Wijaya, C. Y. (2020, September 14). *5 SMOTE Techniques for Oversampling your Imbalance Data*. Retrieved from Towards Data Science: https://towardsdatascience.com/5-smote-techniques-for-oversampling-your-imbalance-data-b8155bdbe2b5

World Health Organisation. (2019, June 14). *Drinking-water*. Retrieved from World Health Organisation: https://www.who.int/news-room/fact-sheets/detail/drinking-water

Yiu, T. (2019, June 12). *Understanding Random Forest*. Retrieved from Towards Data Science: https://towardsdatascience.com/understanding-random-forest-58381e0602d2