Report

Room-Occupancy-Estimation-Analysis

Mohammad Al Hakawati

## Introduction:

* Describe the problem you are addressing and why is it important?

Problem: This project goal is to address the problem of unnecessary energy consumption used in heating and ventilation, through estimating or predicting the number of occupants in a room relying on non-intrusive sensors’ data such as temperature, light, sound, motion, and CO2 sensors.

Importance: Determining the number of occupants in the room enable us to control the energy amount spent on heating and ventilation, because there is no need to spend energy while there are no occupants in the room, even though there are occupants, the amount of energy differs according to the number of occupants in there. Knowing the number of occupants allow for better and efficient energy consumption, thus, reducing costs, saving energy, and protecting the environment. Furthermore, determining the exact amount of energy to be spent on heating and ventilation reflects on better comfort for the occupants through making the environment more appropriate for them.

* Describe the dataset's source, collection method, attributes, size, and domain.

Dataset source: The dataset was published on UCI Machine Learning Repository, the publishers collected this data and published it alongside a research paper that explains the whole process of collecting data from sensors, implementing machine learning algorithms, and finally, predicting the room occupancy count.

Collection method: The data was collected using a wireless sensor network that was placed in a 4.6m \*6m room, which consists of 7 nodes configured as a star shape all connected to a master node. 5 different types of sensors were used, each node from nodes 1-4 (each one placed on a desk) had a temperature, light, and sound sensor. While node 5, which is in the centre of the room had a CO2 sensor, lastly, each of nodes 6 and 7 had passive infrared sensor (PIR). The data was collected over 7 days, each 30 seconds, the sensors transmit data to the master node through an Arduino Uno microcontroller that is connected to each node, so the difference between observations in the same day is 30 seconds each.

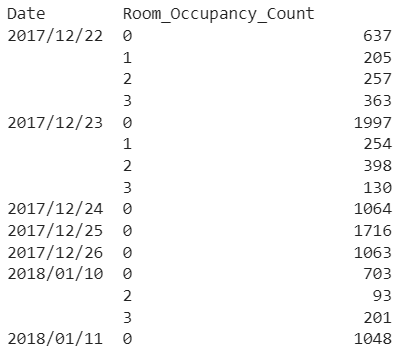
Attributes: The dataset contains 18 attributes, starting with the date and time columns which represent the exact time the observations were collected, followed by 4 columns for the 4 temperature sensors placed on each desk nodes along with the other 4 light and 4 sound sensors, in addition to the 2 PIR sensor columns coming from node 6 and 7.

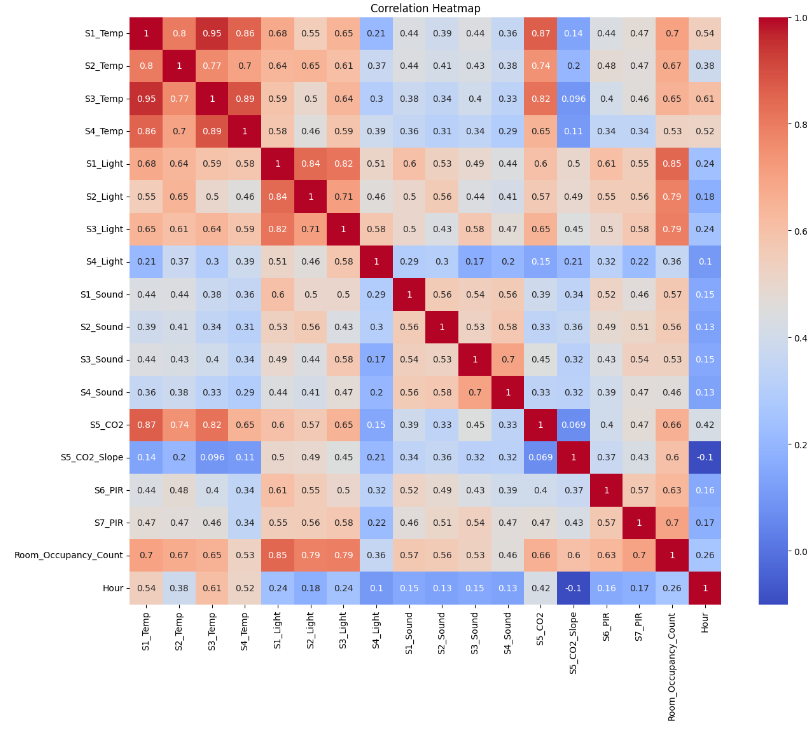
Size: The dataset size is (19,10129), the 19 columns are the 18 attributes mentioned above in addition to the target column, which is the room occupancy count, and 10129 observations.

Domain: The domain of the dataset is energy consumption; it was collected in order to view the room occupancy count and control the amount of energy consumed.

* Describe the learning problem you are trying to solve.

I am trying to predict the room occupancy count and categorize them in a one of the classes from 0-3 (multiclass classification problem), through implementing 4 classification algorithms which are Random Forests, SVM, Gradient Boosting, and XG Boost, each algorithm will be implemented multiple times, first, the data without scaling and hyperparameter tuning, followed by the same model but with scaled version of the data, and finishing with grid search for hyperparameter tuning and the scaled data as the training data, after following this process for all of the 4 algorithms, an evaluation for the results will be held through some visualization. For further understanding for the dataset, some exploratory data analytics will be applied, in addition to performing the required pre-processing before implementing the algorithms.

* How did you prepare training and test data before implementing machine learning models?
* Checking the sum of null values in each column in order to handle them if any.
* To explore the target variable, I reviewed the count of each class which led me to the conclusion that the data is imbalanced with class 0 is the dominant class, while all other classes values were closer to each other.
* In order to gain more information about the data, especially the target variable, I used ‘group by’ to review how the response values are distributed in each date. From the result shown below, there are 4 days that only contained observations with 0 occupants, while in the other 3 days, the number of occupants varied and had values other than 0. So, I dropped the date column as it is irrelevant and doesn’t give additional insights.
* A screenshot of a computer

  Description automatically generatedTo view the difference between the reading of the PIR sensors on node 6 and 7, I grouped them together along with the target to see how both of them detect motion, from the figure below, we can notice that most of the times when both of them return 0, there number of occupants=0, while when both of them return 1, almost all times the number of room occupants is anything but 0.
* As mentioned above, the model’s will be trained on scaled and original data, so I scaled x after splitting the dataset into x and y, and I saved the values of the scaled data into a variable called x\_stan.
* To deal with the time in an easier way, I extracted 3 features which are the hour, minute, and second. And To determine whether I should drop more columns or not, I used feature importance which measures each feature importance in predicting the response. From the obtained results, I decided to drop the minute and second columns I extracted from the time column, as their importance is very small compared to other features, and because I wanted my project to focus on the sensor data in predicting the room occupancy count rather than other features.
* ****I used the correlation matrix, which is represented in the heatmap below, to show the correlation between variables that will be used in predicting the response after dropping all unnecessary features, especially between the attributes and the response, the light sensors have the highest correlation with the response, followed by the temperature, PIR, and CO2 sensors. Also, it shows that there is a linear relationship between the features and the label.

### Methods*:*

* Explain why the provided models are appropriate to solve this problem.
* The provided models are suitable for the room occupancy estimation problem due to their ability for classification tasks. Moreover, these models are robust and capable to handle imbalanced data, as shown in the data uneven distribution across different occupancy counts.
* Starting with Random Forest, which is recognized with its robustness to the outliers and can deal and capture patterns even in minority classes such as our dataset. Although, RF allow assigning weights to classes, which is used with the k-fold cross-validation.
* Moving to SVM, where it excels in high-dimensional spaces, as long as in capturing linear relationship classification, like our dataset. Moreover, it works well with medium to small datasets and can capture the relationships and patterns well.
* The ensemble methods, Gradient boosting and XGBoost, are beneficial for such classification problems where the dataset is small and imbalanced. These methods excel by making a series of weak learners to create a robust predictive learner that works well with the training data and also generalize well to the new data.
* All these provided models is expected to work effectively with the problem and dataset, considering imbalanced and small-sized dataset. Moreover, using grid search with k-fold cross-validation allow for hyperparameter tuning, which will ensure to get the optimal performance for each model.
* Demonstrate how you will test the machine learning application using a range of test data and explain each stage of this activity (Apply k-fold cross-validation).

In order to accurately test the machine learning model on the test data, several methods were implemented.

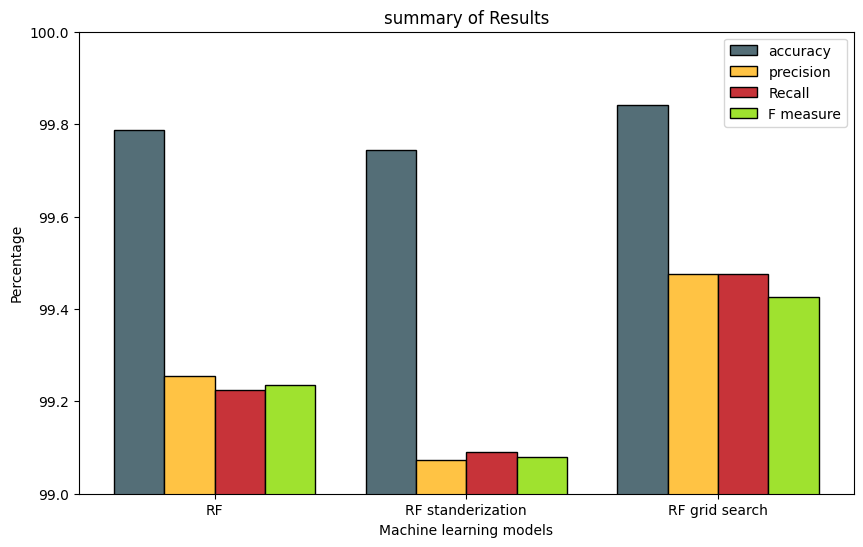
* Firstly, conducting iterations (10 iterations) on each model, evaluating their performance on standardized and non-standardized data each. This method was to ensure the reliability and consistency of models’ evaluation. This iterative approach served in capturing variations in training data, showing each model performance in different conditions.
* After this approach, a grid-search was used to tune the hyperparameters. This approach led to identify the most optimal combination of hyperparameters, enhancing the models’ predictive performance.
* For more robust evaluation and enhanced generalization, k-fold cross-validation were used. The dataset was partitioned into k subsets, and each model was trained and tested on different folds, allowing for a thorough assessment of the models' performance across distinct subsets and a more accurate estimation of their generalization capabilities.
* Explain in detail the machine learning algorithms you are using to address this problem.
* **Random forest:** an ensemble method that combines multiple trees in randomly selected data subsets and take majority vote of them to decide the class. These trees are split using a random subset of the predictors from the full data predictors, where each tree is trained on a different subset of the data. This diversity ensures to create a model that is more robust, generalize well on unseen data and not overfitted.
* **Support vector machine (SVM):** a supervised machine learning model works by drawing a hyperplane that best separates the data between the different classes and maximizing the margin between them. SVMs excels in high-dimensional spaces and well suited for small to medium sized data. In some cases, the data isn’t linearly separatable and need to add more features, which can result in linearly separable dataset. But in SVM, a technique can be done called Kernal trick which transform data into higher dimensional space. By using this technique, the non-linearly separable data is converted into linearly separable one.
* **Gradient Boosting:** involves a sequential creation of weak learners, which are often decision trees, in order to fix errors from earlier trees. The goal of the procedure is to reduce residual errors from earlier trees. Each tree's contribution is determined by its learning rate; a greater rate leads to fewer trees for quicker convergence. However, this strategy may result in a less robust model. Aggregation combines the initial tree prediction with a scaled version of the subsequent trees.
* **XGBoost:** is a refined version of Gradient Boosting that excels in performance and speed. It adds a regularization that penalize adding extra tree to the model using L1 and L2 norm. Additionally, XGBoost builds trees to their maximum depth first, then prunes weak branches in the reverse direction. Moreover, even it is a sequential algorithm, it parallelizes the decision tree building process to fasten the process.

### Evaluation*:*

* What performance measures did you use to evaluate the effectiveness of your models?
* **Accuracy:** is calculated as the sum of true positives and negatives divided by the total number of samples. It shows how accurate the model's predictions are for every class.
* **Recall**: Measured as true positives divided by the total of true positives and false negatives, recall indicates to the ability of a model to identify important instances within the data.
* **Precision**: is calculated as true positives divided by the total of true positives and false positives. It indicates how well the model predicts positive outcomes and highlights how well it avoids false positives.
* **F1-Score:** it is the harmonic mean between the recall and precision. it provides a single statistic that considers false positives and false negatives. providing an inclusive assessment of the model's performance.
* Why did you use these metrics?
* The use of these particular metrics: accuracy, recall, precision, and F1-Score was driven by their suitability for evaluating classification problems – like ours. Where accuracy indicates to how well the model predicts correctly in general, but in our problem and dataset, we shouldn’t demand on the accuracy because the data unbalanced so the accuracy will be biased in predicting the major class. Where other measures are suitable to be taken into consideration as they consider the whole classes.
* The recall can be focused on when we want to see how well the model captures the important instances. Where precision, is when we want to avoid the model to predict the false positives. Or we can focus on a holistic measure that considers both false positives and false negatives.
* Evaluate how, based on the performance measures, you were able to enhance the model.
* Firstly, after using the baseline of each model on the data then calculating the performance measures, I decided to scale the data and train the model again. After reviewing the increase in the performance measures, I decided to use the standardized data as it improves my models’ performance especially in the SVM.
* Secondly, after seeing the importance’s of each feature and the performance measures of each model. I decided to drop them, as some of them affects the models whether negatively or neutrally. So, reducing the models’ complexities and improving performance measures in some models were the reasons behind dropping columns.
* Thirdly, I used the class weights hyper parameters in the models, as it affected the performance positively and lead to better models.
* Lastly, based on the performance measures, I tuned the hyper parameters especially in the grid search. All of these approaches have improved the model performance significantly and reduced the complexity of each model.

### Results and Discussion*:*

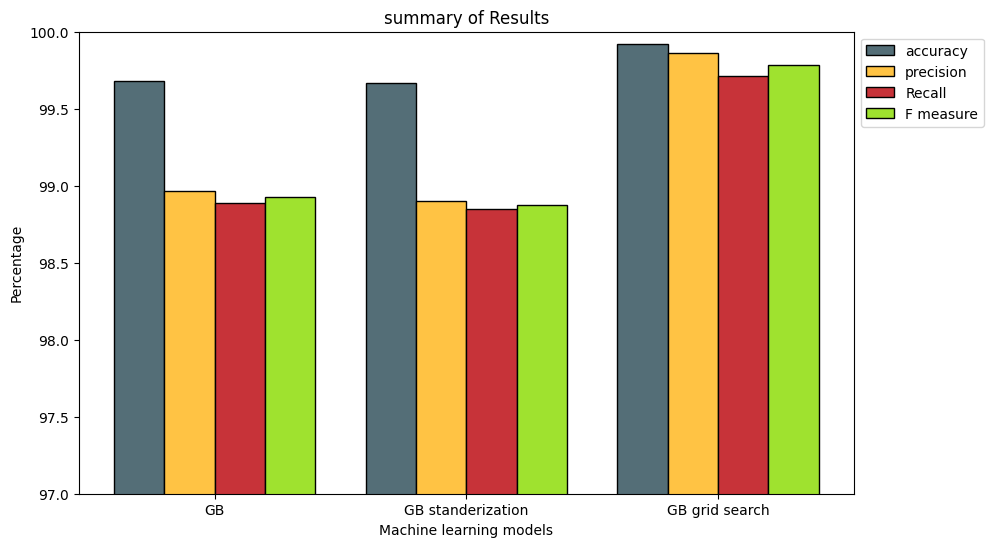
* Discuss the reliability of your results and whether they are balanced, overfitting, or underfitting.
* The results are reliable since the performance metrics on unseen test data have balanced values. Metrics such as F1-Score, Accuracy, Precision, and Recall showed consistent and acceptable performance across the machine learning models. This balance reduces the possibility of overfitting or underfitting by indicating that the models generalize effectively to new, untested data. The performance measures on the test data would differ greatly and be much worse from those seen during training if overfitting or underfitting were existent.
* Additionally, the reliability of the results is strengthened by a careful approach to model training and assessment. Multiple iterations on both initial models on standardized and non-standardized data provided an extensive overview of model behaviour under different conditions. Furthermore, the findings are more reliable when grid search with k-fold cross-validation is used during hyperparameter tuning. The results are more reliable because of this methodical approach.
* Analyse the result of the applications to determine the effectiveness of the algorithms.

This plot shows the difference in the performance measures of Random Forest model among non-standardized, standardized and with grid search. We can see that random forest with the standardized data perform worse. Where is the best model was after hyperparameters tuned in the grid search with k-fold cross validation.

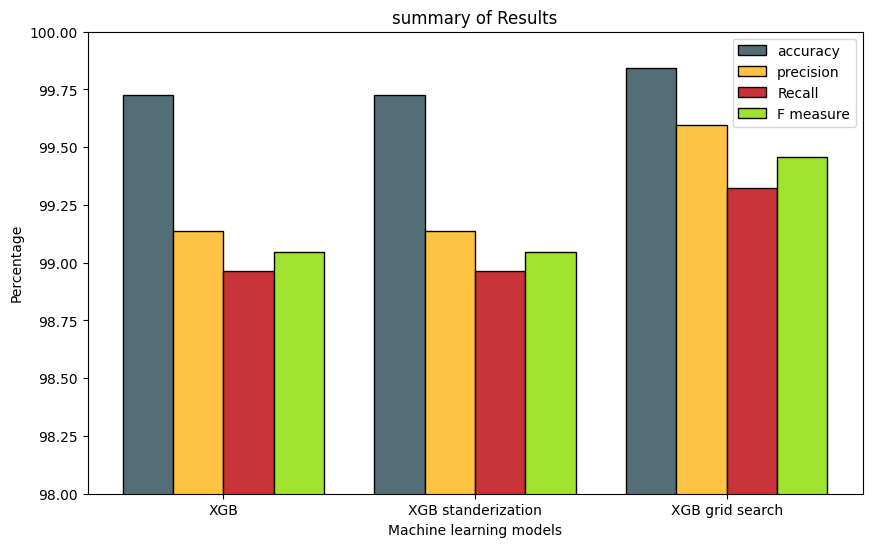
A graph of different colored bars

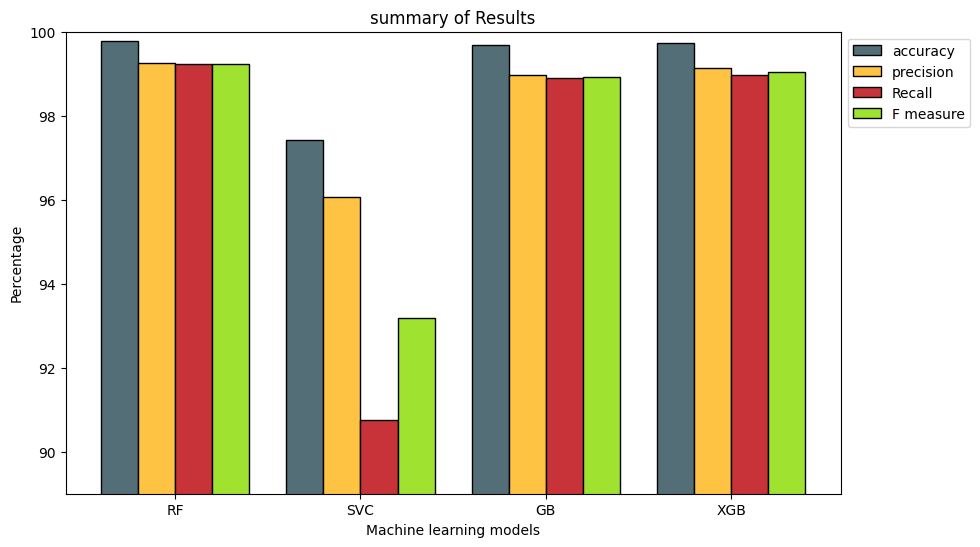
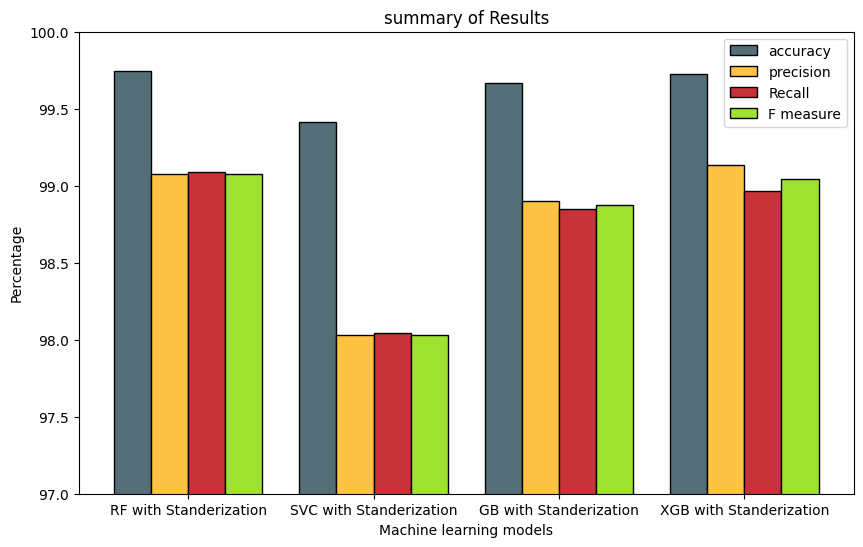
Description automatically generated

This plot shows the SVC among non-standardized, standardized and with grid search. We can determine that SVC performed significantly better after the data was standardized as it is sensitive to the data standardization. Moreover, the model performed the best after the grid search with standardized data.



In this graph is the difference in the Gradient Boosting. The model evaluation metrics didn’t change after using standardized data. But, after the grid search the model performance increased and performed much better.

As in the Gradient Boosting, the XGBoost performance measures didn’t change after standardizing the data, but the performance increased after the model was used in grid search with K-fold cross validation. Which means that the hyper parameters were tuned better.

This plot demonstrates the base models when trained on the data before standardization. We can figure out that SVC was the worst model among all measurements. The other three models were close in the measures, but the RF was the best among all models. Which indicates that RF was the best in detecting the patterns in data before standardization.

In this plot we draw the models’ performance on the standardized data among all models. Even the SVC was significantly performed better, the other models still fitted and predicted the data better. The other three models; measures were close, where is the best model was the RF taking into consideration all measures.

A graph of different colored bars

Description automatically generated with medium confidencethis plot shows the performance of each model after hyper parameters in the grid search. We can figure out that models’ performance differentiated after the grid search. The Boosting algorithms performance have increased especially the GB achieved the best performance among all models after hyper parameters were tuned.

* Draw conclusions regarding the strengths and weaknesses of the different algorithms?

The initial action that is required in Support Vector Classifier, as after standardization it proved to be effective was standardizing the data before using the SVC. By addressing changes in feature scales, this preprocessing approach improved the performance of the model. Standardization made guaranteed that every feature affected the algorithm in the same way, which increased predicted reliability and precision.

Despite its strength in hyperparameter tuning, the grid search approach has significant computing power. It can take a lot of time and resources to fully try hyperparameter spaces, especially when using ensemble techniques like boosting algorithms. This limitation emphasizes the need for caution when using grid search, particularly with algorithms that need a significant amount of computing power.

Strong generalization was shown by the Random Forest's default parameters on test data, demonstrating the algorithm's fundamental robustness. This means that in some circumstances, RF's default parameters are appropriate for the job and even beneficial because they enable good outcomes without requiring plenty of hyperparameter tuning.

Boosting methods, such XGBoost or Gradient Boosting, demonstrated improved performance in capturing complicated data relationships after grid search. One important benefit is that boosting algorithms can gain a great deal from hyperparameter tuning. The grid search increased the predictive performance of these algorithms by making it easier to identify the best hyperparameter.

* Identify further enhancements which can be done in the future? ﻿Discuss any limitations and future improvements of your project.
* Some algorithms’ performance might differ based on the hyperparameters assigned values, so, further tuning for hyperparameters might enhance some algorithms’ performance.
* Check if the dataset contains outliers and handle them in based on the nature of the data, and their influence on the performance of the implemented algorithms.
* From the EDA applied, I discovered that the data is imbalanced, balancing the dataset using overfitting or underfitting instead of class weights might impact the performance of the algorithms positively.
* As it was shown on the heat map in the introduction section, I observed that some sensors have low correlation with the target which might decrease the models’ performance, performing feature selection might remove these features resulting in a better and less complex model.
* In the future, we can do feature extraction such as what they did in the data when they extracted the CO2 slop feature out of the CO2 feature, such new feature may improve the overall results.
* One of the limitations that I faced in the project is the high computational power the algorithms require, especially with the iterations I added, which also resulted in an increased runtime for the code and the algorithms.

## References

*What is Random Forest?* (no date) *IBM*. Available at: https://www.ibm.com/topics/random-forest (Accessed: 28 January 2024).

Gandhi, R. (2018) *Support Vector Machine - introduction to machine learning algorithms*, *Medium*. Available at: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47 (Accessed: 28 January 2024).

Saini, A. (2024) *Gradient boosting algorithm: A complete guide for beginners*, *Analytics Vidhya*. Available at: https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/ (Accessed: 28 January 2024).