Machine learning – Final Assignment

Name	Hamza Muhsen
Class	2
Assignment title	Assignment 2 ML Project
Date	30-1-2024

1. Introduction:

- **Describe the problem you are addressing and why is it important?**
  - The problem in [1], was the large amounts of energy consumed from the heating, ventilation, and air conditioning systems.
  - This large amount of energy can negatively affect the environment by contributing to carbon emissions spread which impact humans' health, animals, etc.
  - The focus is to optimize the energy consumption from heating, ventilation, and air conditioning systems and save energy by using machine learning models and multiple functions sensors to implement and build heating, ventilation, and air conditioning systems demand-driven.

- **Describe the dataset's source, collection method, attributes, size, and domain**.
  - The paper's [1], aim was to decrease the energy consumed from HVAC systems by using machine learning.
  - This problem falls comes under the buildings and smart buildings focusing on optimize the energy consumed from HVAC systems and decrease the spread carbon emissions resulted from HVAC systems.
  - They collected the data by using a room and a star configuration with seven non-intrusive sensors to measure and calculate the environmental parameters such as light, temperature, sound, CO2, and motion for four days.
  - These parameters helped them to calculate and define room occupancy which is the target in their study.
  - The dataset contains 10129 instances and 18 features to solve the problem and implement machine learning algorithms.
- **Describe the learning problem you are trying to solve.**
  - My job is to accurately define room occupancy by using data collected (sensors) and machine learning algorithms.
  - The problem is a classification problem because the target variable is categorical data
  - The selected machine learning algorithms are: random forest, SVM, gradient boost , and XGBoost.
- **How did you prepare training and test data before implementing machine learning models?**
  - Import dataset and gain some information about it by showing datatypes and statistical summary.
  - Check nulls in our data but there weren't any nulls in the data.

- Apply normalization using MinMax scaler on the features expect the target variables and date. All the features after normalization have the same range (from 0 to 1).
- Correlation between features and apply feature selection based on correlation between features and with target variable. If the correlation between two features is greater than 80, then one feature must remove, so choose the feature with bigger correlation with target variable and keep it while the remove the another.
- Define X and y → X all features after feature selection expect the target variable “Room occupancy Count”, and Y is the target variable “Room occupancy Count”.
- Split data into training and testing sets and define the test set size (30%).
- In our data, the data was imbalanced, so I handled the imbalanced data using SMOTE technique.

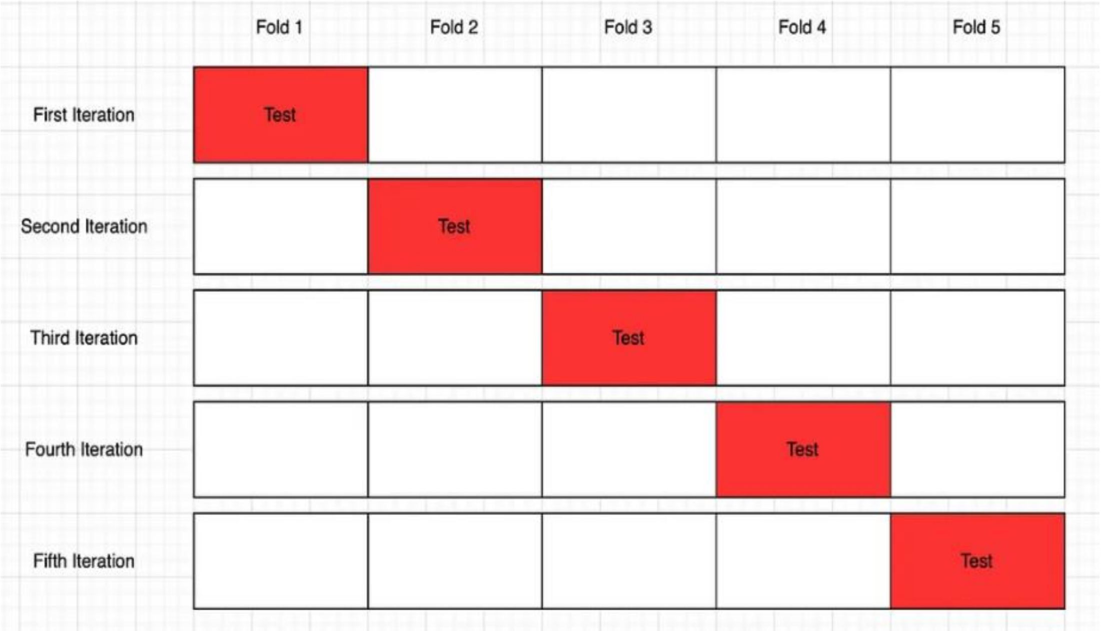
2. Methods*:*

- **Explain why the provided models are appropriate to solve this problem.**
  - All machine learning algorithms selected was based on the problem which is considered as classification problem.
  - Random forest, SVM, gradient boost and XGboost are effective and provide accurate outcomes and predictions specific to classification problems.
  - With the use and help of characteristics of each selected models such as : ensemble learning in random forest to deal with non-linear relationships and overfitting , gradient boost is also a very good choice to deal with complex data and has the ability to detect complex relationships between features like our dataset, SVM and its ability to deal with high-dimensional data which is a good choice specially in our data.

- **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).**
  - Min max scaler
    - It is a normalization technique implemented in machine learning. It transforms features to be within the same range between 0 and 1. It helps the algorithms to provide better performance.
    - The equation of minmax:

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

  - SMOTE
    - It is the technique used to handle the imbalanced class distribution in machine learning.
    - The goal of it to ensure that the machine learning model has no bias toward any one class by creating an equal number of the classes.
    - Every minority class instance is chosen by SMOTE along with its k nearest neighbours. In addition, the line segments linking the minority instance with the other (neighbors).
    - The process will repeat until reach the balance between majority and minority.
      (Satpathy, Smote for imbalanced classification 2023) [2]
  - Grid search with Cross-validation
    - It is considered one of the hyperparameters tuning techniques used to search and explore using a model's predetermined group of hyperparameter values.
    - The goal of grid search is to find the best hyperparameter for specific model to perform well and get the best performance.
    - Hyperparameter space → it is a list of hyperparameters and their values that you prefer to explore. After that, train model for each combination of parameters in the grid.
    - Evaluate model performance by using cross validation on the training set.
    - Cross validation requires splitting the dataset into multiple folds, and then train the model on majority of folds and test on the remaining folds.
    - The process above is repeated for each fold being held as the test group. The resulting model is calculated using the average of the models.
    - One of the most popular methods of cross validation is “k-fold” cross validation. (k is the number of folds). k=5 is one of the most used.

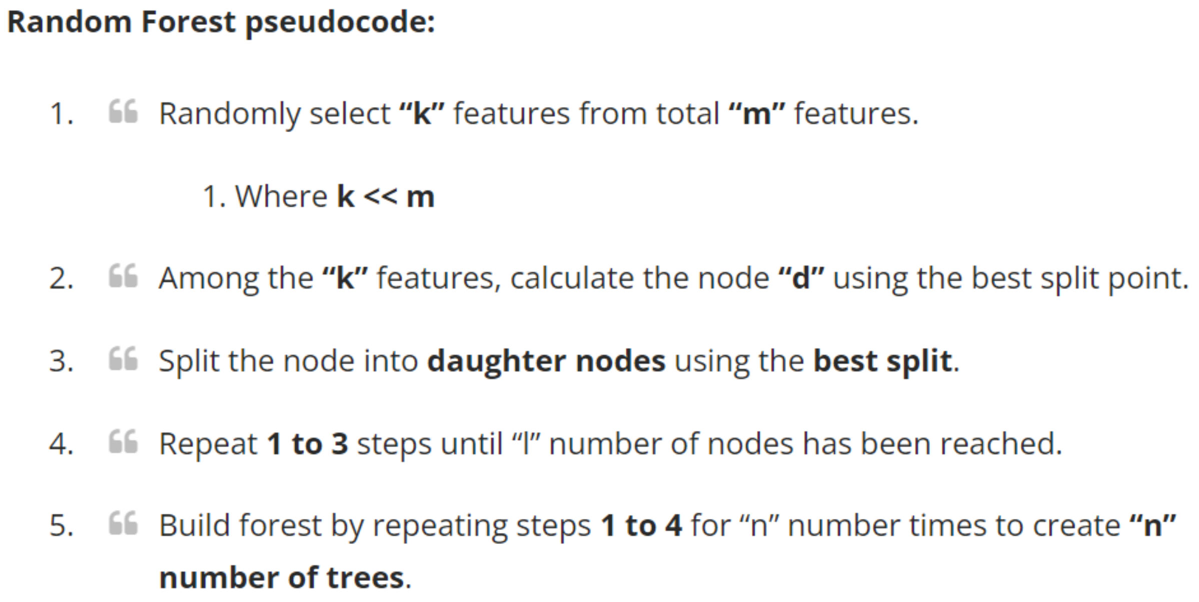


	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
First Iteration	Test				
Second Iteration		Test			
Third Iteration			Test		
Fourth Iteration				Test	
Fifth Iteration					Test

- As shown in the figure above, there is 5 folds, and each fold contains 4 training and 1 test. The final cross-validation model is created by averaging the results of the five rounds after they are finished.
- Implement a performance metric to evaluate and assess model performance.
- After than based on evaluation metrics, select the best performance on the test/validation data.

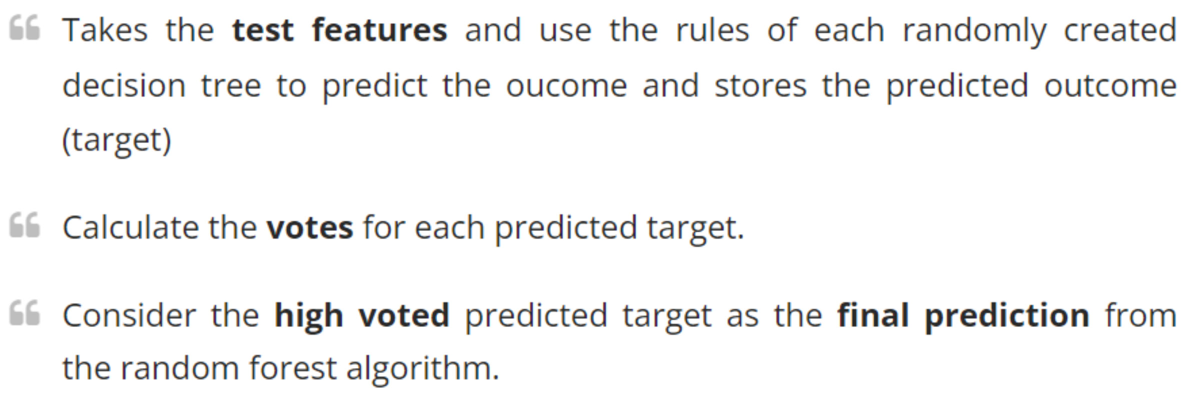
(Beheshti, Cross validation and grid search 2022) [3]

- **Explain in detail the machine learning algorithms you are using to address this problem.**
    - Random Forest
        - It is a combination of ensemble methods and decision tree model to build and create various decision trees from your data.
        - Bagging → during the training, each decision tree uses subset of training data that is different from each training data. All this achieved by using bootstrap. Each decision tree learns different sector or aspect of the data.
        - Voting → each decision tree gives its own prediction, after that, the final prediction produced by determining the majority vote.
        - Randon forest can solve classification and regression problems.
        - It handles non-linear relationships and overfitting.
        - n_estimators (number of trees), max_depth (maximum depth f each tree) ,min_samples split (minimum number of samples necessary for splitting an internal node) are considered the main hyperparameter in random forest.



Random Forest pseudocode:

1. Randomly select **“k”** features from total **“m”** features.
    1. Where **k << m**
2. Among the **“k”** features, calculate the node **“d”** using the best split point.
3. Split the node into **daughter nodes** using the **best split**.
4. Repeat **1 to 3** steps until “l” number of nodes has been reached.
5. Build forest by repeating steps **1 to 4** for “n” number times to create **“n” number of trees**.



Takes the **test features** and use the rules of each randomly created decision tree to predict the oucome and stores the predicted outcome (target)

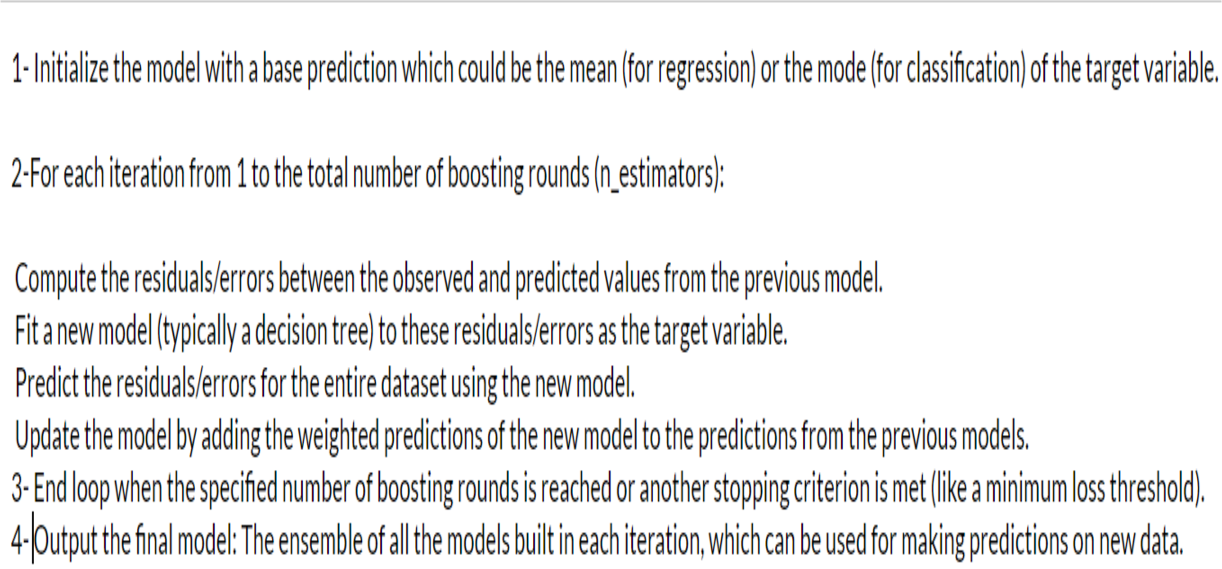
Calculate the **votes** for each predicted target.

Consider the **high voted** predicted target as the **final prediction** from the random forest algorithm.

(Beheshti, Random Forest Classification 2022) [4]

(How the Random Forest algorithm works in machine learning 2020) [5]

- Gradient Boost
  - It is considered on of the boosting algorithms that enhance the performance of the model and prediction. It is an ensemble learning that generates a robust and effective prediction model through merging predictions from multiple weak learners (usually decision tree).
  - Every weak learner fixes the error made by the one before it to reduce the residual error of the previous one.
  - Gradient descent implemented in gradient boost to minimize the loss by continuously upgrading the model.
  - Small value of learning rate gives large numbers of decision trees and better performance than higher value of learning rate. On the other hand, lower value of learning rate increases the time.
  - Higher value of learning rate increases the process of learning but with possibility to get overfitting.
  - Combining involves including the first tree prediction plus a scaled (shrunk) version of the subsequent trees.
  - Learning rate is one of the most important hyperparameter in gradient boost so it’s important to find the best value to fit the data. In our case, I used grid search to find the best learning rate.
  - GB is generating accurate models, deal with all datatypes and implement in many sectors and applications.
  - On the other hand, it requires more attention during hyperparameter tuning specially learning rate. Wrong hyperparameter tuning can lead to overfitting and other challenges.



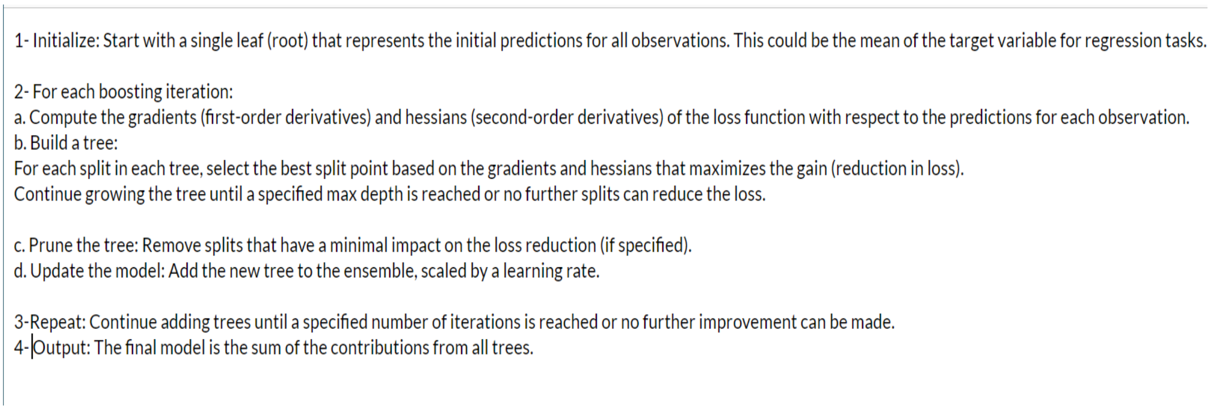
1- Initialize the model with a base prediction which could be the mean (for regression) or the mode (for classification) of the target variable.

2-For each iteration from 1 to the total number of boosting rounds (n_estimators):

Compute the residuals/errors between the observed and predicted values from the previous model.
Fit a new model (typically a decision tree) to these residuals/errors as the target variable.
Predict the residuals/errors for the entire dataset using the new model.
Update the model by adding the weighted predictions of the new model to the predictions from the previous models.
3- End loop when the specified number of boosting rounds is reached or another stopping criterion is met (like a minimum loss threshold).
4- Output the final model: The ensemble of all the models built in each iteration, which can be used for making predictions on new data.

(Zaburo, Gradient boosting と xgboost 2017)[6]

- XGBoost
  - Like the gradient boost, XGBoost belongs to the gradient boosting frameworks.
  - XGBoost is an updated version of gradient boost but with higher performance and speed in multiple machine learning problems.
  - Unlike gradient boost, XGBoost includes regularization techniques to handle and overcome overfitting problem.
  - In addition, XGboost use cross validation automatically in each tree to get rid of overfitting, deal with missing data.
  - It also runs the gradient boost calculations in the CPU cache memory which is the reason of reducing time needed to run model.
  - It can handle and apply on complex and large data because it supports parallel computing.
  - Like gradient boost, XGBoost requires more attention in hyperparameter tuning and select the useful hyperparameters that enhance and get the best performance.
  - Furthermore, it is sensitive to outliers and with outliers it can’t provide the best performance.



1- Initialize: Start with a single leaf (root) that represents the initial predictions for all observations. This could be the mean of the target variable for regression tasks.

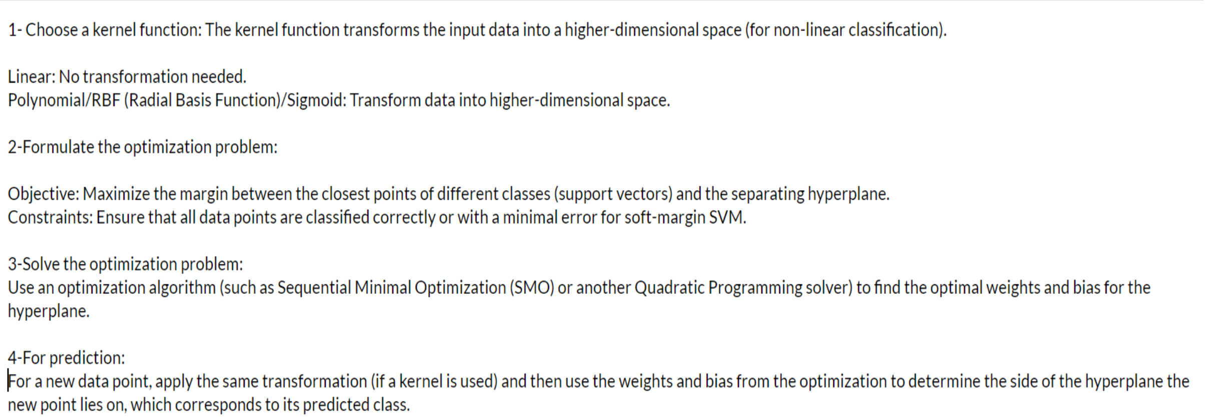
2- For each boosting iteration:
a. Compute the gradients (first-order derivatives) and hessians (second-order derivatives) of the loss function with respect to the predictions for each observation.
b. Build a tree:
For each split in each tree, select the best split point based on the gradients and hessians that maximizes the gain (reduction in loss).
Continue growing the tree until a specified max depth is reached or no further splits can reduce the loss.

c. Prune the tree: Remove splits that have a minimal impact on the loss reduction (if specified).
d. Update the model: Add the new tree to the ensemble, scaled by a learning rate.

3-Repeat: Continue adding trees until a specified number of iterations is reached or no further improvement can be made.
4- Output: The final model is the sum of the contributions from all trees.

(Zaburo, Gradient boosting と xgboost 2017)[6]

- SVM
  - Abbreviation of Support Vector Machine.
  - Supervised machine learning algorithm used with classification and regression.
  - Recommended and suited to use for high dimensional spaces(number of features > number of samples), complex , small and medium data.
  - It has the ability to find the best hyperplane that perfectly separates different classes.
  - There is something called “Margin” which is the distance from hyperplane and datapoint (nearest) from each class.
  - It transforms the input into higher dimensional spaces which make it difficult to find hyperplane that perfectly separates classes which called “Kernel trick”. Kernels functions are like linear, poly, and RBF.
  - C parameter handles the trade-off with providing a smooth decision boundary and properly categorizing training points. (small C → more misclassification), (larger C → correct classification).
  - SVM can be used in binary (divide data point → two classes, and hyperplane is applied to maximize “margin” between two classes)or multi class classification(by using method like one vs the rest).
  - SVM is also sensitive to outliers which may impact the position of the hyperplane.



1- Choose a kernel function: The kernel function transforms the input data into a higher-dimensional space (for non-linear classification).

Linear: No transformation needed.
Polynomial/RBF (Radial Basis Function)/Sigmoid: Transform data into higher-dimensional space.

2-Formulate the optimization problem:

Objective: Maximize the margin between the closest points of different classes (support vectors) and the separating hyperplane.
Constraints: Ensure that all data points are classified correctly or with a minimal error for soft-margin SVM.

3-Solve the optimization problem:
Use an optimization algorithm (such as Sequential Minimal Optimization (SMO) or another Quadratic Programming solver) to find the optimal weights and bias for the hyperplane.

4-For prediction:
For a new data point, apply the same transformation (if a kernel is used) and then use the weights and bias from the optimization to determine the side of the hyperplane the new point lies on, which corresponds to its predicted class.

(Saini, Guide on Support Vector Machine (SVM) algorithm 2024)[7]

3. Evaluation*:*

Evaluate the effectiveness of the learning algorithms used by answering the following questions:

- What performance measures did you use to evaluate the effectiveness of your models?
  - Accuracy → describes and shows the overall correctness of the model by using this equation:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{All Samples}}$$

Accuracy gives us the ability to general view of our model performance. Higher accuracy means higher performance and prediction.

- Recall → the model ability to define all positive instances. We can use it to see the percentage of positives well predicted.

$$recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

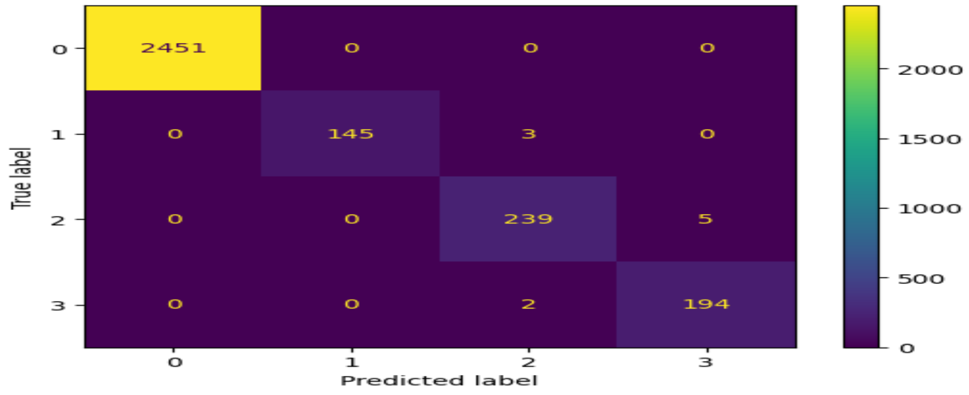
- Precision → it displays the number positive predictions well made, higher value of precision, our model will more minimize the number of false negative.

$$precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

- F1 score → combination of precision and recall and it’s providing us with the ability to see a good evaluation (performance) of our model. It uses the harmonic mean to calculate the average of recall and precision.

$$\text{F1 Score} = \frac{2 \text{ x (Precision x Recall)}}{\text{Precision + Recall}}$$

- **Confusion matrix** → it is a table shows the model predictions and actual data (outcomes).



(Agrawal, Metrics to evaluate your classification model to take the right decisions 2023) [8]

- Why did you use these metrics?
  - These metrics were chose based on the problem defined which is classification problem not regression. All metrics used are the most popular evaluation metrics used to evaluate classification models.

  - Accuracy used in our case because our data is balanced and to get the overall performance of our models. F1 score is also used if we want to focus on negatives over positives, precision (focus on false positive is important, and make sure that positive predictions are reliable).
  - Overall, I used all these metrics to compare between all models selected and define the best model that has the best performance (higher in all evaluation metrics).

- Evaluate how, based on the performance measures, you were able to enhance the model.
  - For me, I used the performance metrics to compare between models. For example, for the first time fitting random forest, gradient boost, SVM, or XGBoost, I used the default parameters and then used the grid search with different parameters to compare and the model results.
  - I assumed that with the use of grid search must give better performance than default parameters. So, in case that with grid search were less performance than default, I changed and used other parameters in grid search to get better performance.
  - So, by using performance measures, I enhanced the model performance.

4. Results and Discussion*:*

- Discuss the reliability of your results and whether they are balanced, overfitting, or underfitting.
  - From the preprocessing techniques, models used, and evaluation. I think that my results are reliable and accurate because I used:
    - Cross validation → it helps in avoiding the overfitting problem by training and assessing each algorithm on different fold which gives a method to define model's generalization performance.

      Also, by using the cross validation, it helps to define the best hyperparameter to perform well to different subsets of the data and gives more reliable evaluation of the model's generalization. This helped me to prevent overfitting.

    - Grid search → it decreases overfitting by exploring and defining the best parameters to use in the model that enhance the model performance and balance model complexity and generalization.

By doing grid search and select the best hyperparameters for each model used in our case, it helps the model from being overly complex and avoiding choosing hyperparameters that could lead us to overfitting.

  - Complex models → As discussed before, using ensemble methods, and complex models that are handle the overfitting without using grid search or cross validation.

    There are many models like XGBoost that contains regularizations like L1 and L2 to prevent and decrease the overfitting.

    Models I used in this project have an important role to get reliable performance and prediction and avoid overfitting and underfitting.

- Dataset used → In addition, the data played an important role in preventing underfitting and overfitting because the dataset used was perfect and contains large amount of data to enable models to understand patterns variations in the data, which helps in reducing underfitting.

  Also having balanced classes help the model to learn all classes without exception, which help in decreasing underfitting.

- Finally,

All these steps are implemented in my project to give reliable and best performance and avoid any challenges such as overfitting and underfitting.

- Analyse the result of the applications to determine the effectiveness of the algorithms.

Model	Accuracy	F1 score	Precision	recall
Random forest – default	99.67%	0.99	0.99	0.99
Random forest- grid search	99.70%	0.99	0.99	0.99
Gradient boost - default	99.51%	0.98	0.98	0.98
Gradient boost -grid search	99.67%	0.99	0.99	0.99
XGBoost- default	99.54%	0.98	0.98	0.98
XGBoost – grid search	99.61%	0.99	0.99	0.98
SVM- default	99.14%	0.97	0.97	0.98

SVM- grid search	98.91%	0.96	0.97	0.96

  - All the four models selected performed very well on the dataset and provided the best performance.
  - All the models have better performance than the paper’s model (they got maximum accuracy 98.4 and a high F1 score of 0.953).
  - The best model in our project was the random forest with cross validation and grid search with accuracy 99.7, and f1 score, recall precision → 0.99.
  - Random forest was the best due to its ensemble nature that leads to perform the best performance and get better generalization to new data.
  - In addition, random forest has the ability to handle non-linear relationships, and from our visualization of the data, there were non-linear relationships.
  - Random Forest is having the ability of dealing with either categorical and numerical variables. This ability to adapt is especially useful in situations where the data set contains a wide range of feature types.
  - On the other hand, SVM has the lowest performance in our project because in the dataset, there are huge numbers of outliers found and these outliers negatively impact the performance of SVM.
  - Also, maybe I should add more parameters in the grid search to get better performance for all models, but this will consume a lot of time. That’s why I tried to choose the best hyperparameters in each model in the grid search.

- Draw conclusions regarding the strengths and weaknesses of the different algorithms?
  - Random forest gave the highest accuracy, f1 score, precision, and recall among all models. It also gave a high performance even with the default parameters.
  - Grid search and cross validation provided and enhanced machine learning models performance and prediction.
  - Gradient boost with grid search also gave competitive performance.
  - Predictions can be improved by using the sequential learning strategy.
  - XGBoost with grid search gave good performance specially with the regularization to decrease the overfitting.
  - In XGBoost, Precision was marginally reduced by grid search, suggesting a possible trade-off.
  - SVM was the lowest accuracy, f1 score, precision, and recall.
  - SVM was better with default parameters than grid search.
  - SVM didn’t provide the best performance because of outliers in the dataset. “SVM is sensitive”

- Identify further enhancements which can be done in the future? Discuss any limitations and future improvements of your project.
  - Use more parameters in the grid search to get better model performance.

- Use feature engineering to get new features and get better performance. (feature engineering may lead to underfitting)
- Use additional models to implement on the dataset.
- Try the selected models on different datasets and environment to get better idea of our model performance.
- Explore more related papers that implemented machine learning algorithms within the same domain to test our model with them.
- Collect additional data related to the same domain.
- It's possible that not all of the hyperparameter space has been examined by the current hyperparameter setting. The parameters of the model could be further optimized by using advanced tuning techniques or more thorough search strategies.
- For real-time deployment, the effectiveness of the current algorithms might need to be improved. (Test data on real time environment)

5. References

(No date a) Machine learning based estimation of room occupancy using non-intrusive ... Available at: https://www.semanticscholar.org/paper/Machine-Learning-based-Estimation-of-Room-Occupancy-D.-Raj/0e45701d7913ce0b68b0ed466c9e46202a9be859 (Accessed: 30 January 2024). [1]

Satpathy, S. (2023) Smote for imbalanced classification, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2020/10/overcoming-class-imbalance-using-smote-techniques/ (Accessed: 30 January 2024).[2]

Beheshti, N. (2022) Cross validation and grid search, Medium. Available at: https://towardsdatascience.com/cross-validation-and-grid-search-efa64b127c1b (Accessed: 30 January 2024).[3]

Beheshti, N. (2022b) Random Forest Classification, Medium. Available at: https://towardsdatascience.com/random-forest-classification-678e551462f5 (Accessed: 30 January 2024).[4]

How the Random Forest algorithm works in machine learning (2020) Dataaspirant. Available at: https://dataaspirant.com/random-forest-algorithm-machine-learing/ (Accessed: 30 January 2024). [5]

Zaburo (2017) Gradient boosting と xgboost, Blog. Available at: https://zaburo-ch.github.io/post/xgboost/ (Accessed: 30 January 2024). [6]

Saini, A. (2024) Guide on Support Vector Machine (SVM) algorithm, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/10/support-vector-machinessvm-a-complete-guide-for-beginners/ (Accessed: 30 January 2024). [7]

Agrawal, S.K. (2023) Metrics to evaluate your classification model to take the right decisions, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/07/metrics-to-evaluate-your-classification-model-to-take-the-right-decisions/ (Accessed: 30 January 2024). [8]