An-Najah National University

Department of Artificial Intelligence



Final Project

Data Mining and Analysis

Submitted By: Aladdin Husni Odeh

Submitted To: Dr. Anas Toma

Contents

[Table of figures: 3](#_Toc154951250)

[1. Introduction 4](#_Toc154951251)

[1.1. Objectives 4](#_Toc154951252)

[1.2. Dataset overview 4](#_Toc154951253)

[2. Data Preprocessing and Preparing 5](#_Toc154951254)

[2.1. Data cleaning 5](#_Toc154951255)

[2.2. Introducing binary and multi classification features 5](#_Toc154951256)

[2.3. Initial Assessment 6](#_Toc154951257)

[2.4. Preparing multiple data frames for experimenting 9](#_Toc154951258)

[3. Data analysis 10](#_Toc154951259)

[3.1. Position analysis 10](#_Toc154951260)

[3.1.1. Mean and Std aggregation. 10](#_Toc154951261)

[3.1.2. Position differences, cumulative summation and multiplication. 10](#_Toc154951262)

[3.1.3. Distribution of Features by Position. 11](#_Toc154951263)

[3.1.4. Analyze similar buildings. 12](#_Toc154951264)

[3.2. Flattened data Analysis. 13](#_Toc154951265)

[3.2.1. Flattened features analysis. 13](#_Toc154951266)

[3.2.2. Flattened features product analysis. 13](#_Toc154951267)

[3.3. Clustering analysis. 14](#_Toc154951268)

[4. Model training experiments 15](#_Toc154951269)

[4.1. Training Methodology 15](#_Toc154951270)

[4.2. Experimentation 16](#_Toc154951271)

[4.2.1. Linear regression and Random Forest for flattened data and for aggregated data 16](#_Toc154951272)

[4.2.2. Flattened data with outliers-related features. 17](#_Toc154951273)

[4.2.3. Trying all combinations of positions. 18](#_Toc154951274)

[4.2.4. Trying clustering realted features 18](#_Toc154951275)

[4.2.5. Trying PCA to reduce dimensionality. 19](#_Toc154951276)

[4.2.6. Splitting the output using weighted sum for features. 19](#_Toc154951277)

[4.2.6. Using 14 positions with weight correlation with output. 19](#_Toc154951278)

[4.2.7. Using deep learning 20](#_Toc154951279)

[4.2.8. Using rolling window 20](#_Toc154951280)

[4.2.9. Hybrid approach between clustering and rolling windows. 21](#_Toc154951281)

[5. Results analysis 22](#_Toc154951282)

[5.1. Flattened data set 1 analysis 22](#_Toc154951283)

[5.2. Flattened data set 2 group 2 analysis 23](#_Toc154951284)

[5.3. Recommendations 24](#_Toc154951285)

[6. Conclusion 24](#_Toc154951286)

[7. References 24](#_Toc154951287)

# Table of figures:

[Figure 1: Restructured dataset 5](#_Toc154951809)

[Figure 2: Dataset1 Group1 features with classification features. 5](#_Toc154951810)

[Figure 3: Features histogram in both data sets 6](#_Toc154951811)

[Figure 4: Scatter plot between features and Output 6](#_Toc154951812)

[Figure 5: Boxplot for all features in the 2 datasets 7](#_Toc154951813)

[Figure 6: Correlation matrix between features 8](#_Toc154951814)

[Figure 7: Flattened normalized data 9](#_Toc154951815)

[Figure 8: Correlation between aggregated features and the Output 10](#_Toc154951816)

[Figure 9: Correlation between adjacent positions and output 10](#_Toc154951817)

[Figure 10: Boxplot of features per positions 11](#_Toc154951818)

[Figure 11: Similar building analysis 12](#_Toc154951819)

[Figure 12: Flattened feature v.s. Output 13](#_Toc154951820)

[Figure 13: Multiplication of flattened features 13](#_Toc154951821)

[Figure 14: Clustering analysis for D1G1 14](#_Toc154951822)

[Figure 15: Linear regression and Random Forest for flattened data and for aggregated data 16](#_Toc154951823)

[Figure 16: All positions combinations expirment 18](#_Toc154951824)

[Figure 17: Clustering experiment 18](#_Toc154951825)

[Figure 18: PCA experiment 19](#_Toc154951826)

[Figure 19: Splitting the output using weighted sum. 19](#_Toc154951827)

[Figure 20: Using 14 positions with weight correlation with output. 19](#_Toc154951828)

[Figure 21: Rolling windows. 20](#_Toc154951829)

[Figure 22: Code snippet from the final experiment 21](#_Toc154951830)

[Figure 23: Hybrid approach results 22](#_Toc154951831)

# 1. Introduction

## 1.1. Objectives

In this report, we are going to analyze a dataset that shows the strength of a set of buildings, and it shows different spatial features that represent some metrics of the buildings on different positions and sides. The goal is to analyze and understand the relationship between these features on the different positions to build a model that can predict the actual output of the strength.

The actual strength of the buildings wasn’t generated based on these features; it was measured using an expensive device, and the features were measured using a reasonably priced device. So, the goal here is trying to fit these features into a model that can replace the need for the expensive device.

## 1.2. Dataset overview

This is spatial data with a set of features (F1…F5), where each set of features represents a specific position on one side of the building.

The building has two sides, named G1 and G2, and each side has seven positions. There are two datasets with the same structure, and each of them represents a set of 36 different buildings.

The dataset obtained from an unknown resource has 36 records with the following features:

| **Variable Name** | **Role** | **Type** | **Description** |
| --- | --- | --- | --- |
| ID | ID | Integer | From 1 to 36 |
| Output Measured Value | Label | Integer | The strength of the building that we are aiming to predict. |
| Position | Feature | Categorical | The positions of the building from 1 to 7 |
| D{x} G{y} F{z} | Feature | Integer | The main features of the dataset, where each combination of [x = Dataset number], [y = Group ID], and [Z = Feature number], represent a specific feature on a specific side at a specific position in the building |

# 

# 2. Data Preprocessing and Preparing

## 2.1. Data cleaning

The dataset is a very small one with only 36 buildings, so the initial steps was to observe it from the excel file directly. At the beginning, a small restructure has been conducted on the excel file to make it easier for python to load it in a form of data frame. This only includes preparing headers and renaming them with a format of Dx Gy Fz as shown in the image below:

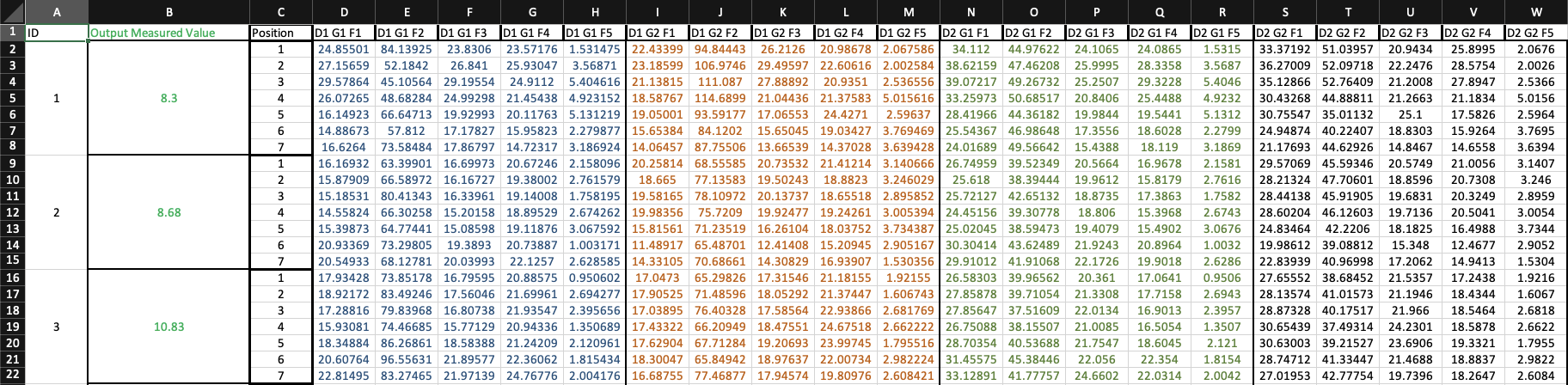


Figure : Restructured dataset

The data had only 1 negative value which was converted to positive since it was clearly a mistake.

## 2.2. Introducing binary and multi classification features

Based on the description of the data, there are no guarantees for any relationship between the features and the label, so 2 new features were introduced for classification experimenting a long side with the regression experimentations.

1. **‘Strength\_2’ feature**: This feature splits the output into 2 categories ‘Weak’ and ‘Standard’. A threshold of 6 was chosen based on an assumption that most probably the data should have more acceptable building compared to defected ones.
2. **‘Strength\_3’ feature**: This feature splits the output into 3 categories ‘Weak’, ‘Standard’, and ‘Great’. Same threshold used to split Weak from other classes, and a threshold of 12.5 was chosen to split ‘Great’ from the rest.



Figure : Dataset1 Group1 features with classification features.

## 2.3. Initial Assessment

The first part of the assessment was to plot a histogram to show the distribution of all features across both data sets as follows:

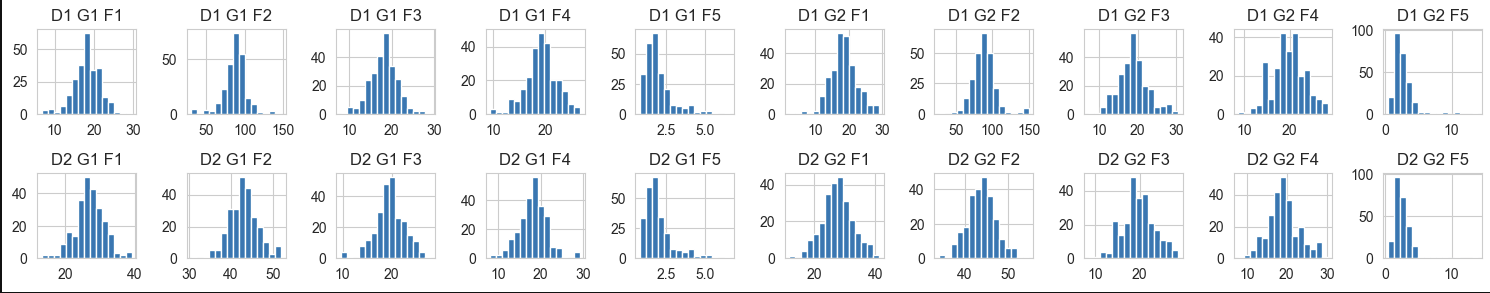


Figure : Features histogram in both data sets

In the first row, we can see all the 10 features (2 group x 5) and in the second row we can see the distribution of the same features in the second data set.

* The histogram shows a very similar distribution between both data sets which is a good start to start experimenting on them separately.
* The first 4 features have a normal distribution compared to F5 which is skewed to the left.

The second part of the assessment was to show scatter plots between each feature in each data set with the ‘Output Measured Value’. The aim of this analysis to understand if there is maybe any direct linear relationship with the output.

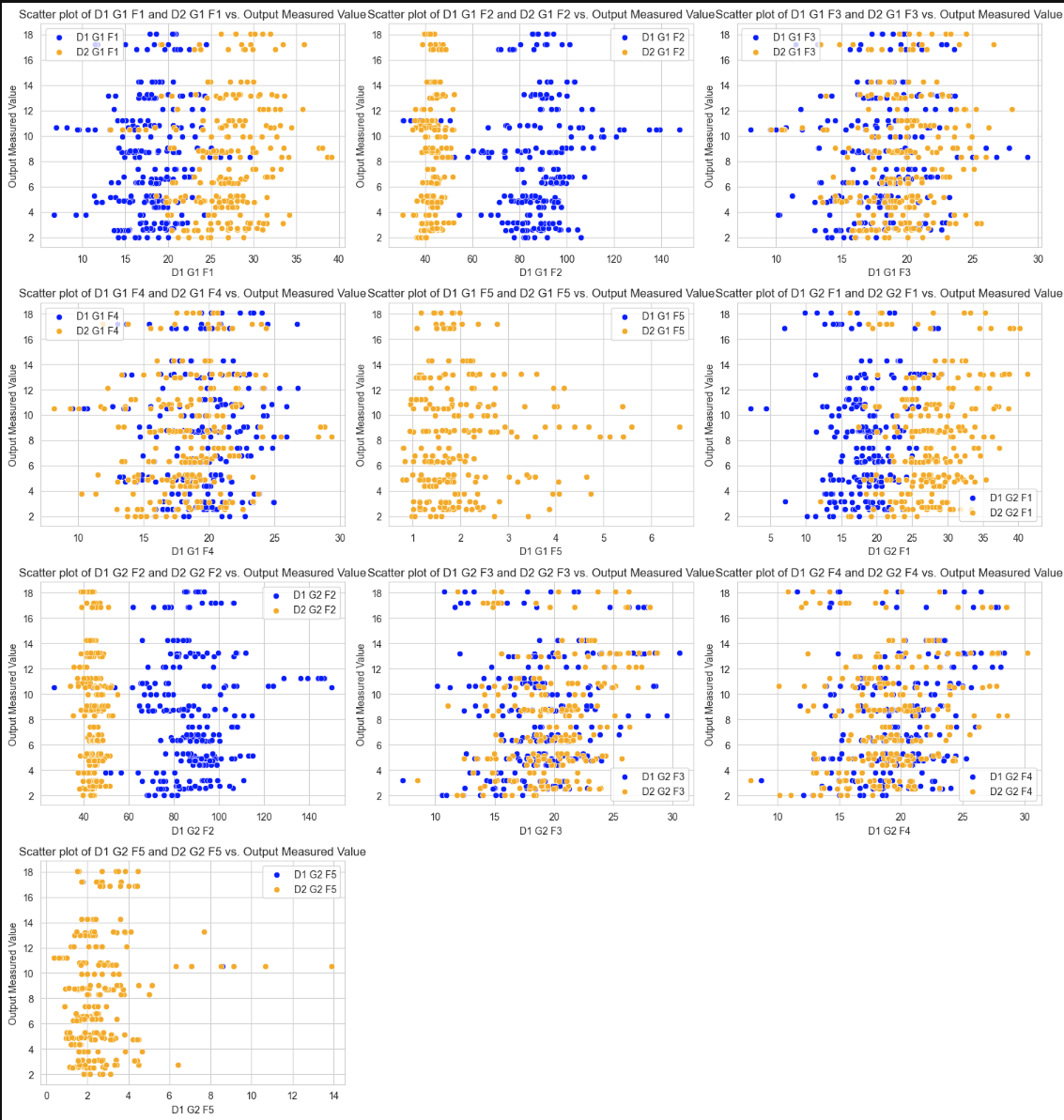


Figure : Scatter plot between features and Output

The results indicate the following:

1. There is no sign of any linear relationship between any of the features with the output which suggests a need of feature engineering to extract complex relationships that could potentially fit the data in a model.
2. F5 in both datasets is identical.
3. The feature in both datasets looks very similar with small differences, mainly some shifts and scaling changes which implies a need for a normalization.

The third step of the initial assessment was plotting a boxplot to understand the ranges of the dataset and to detect any outliers:

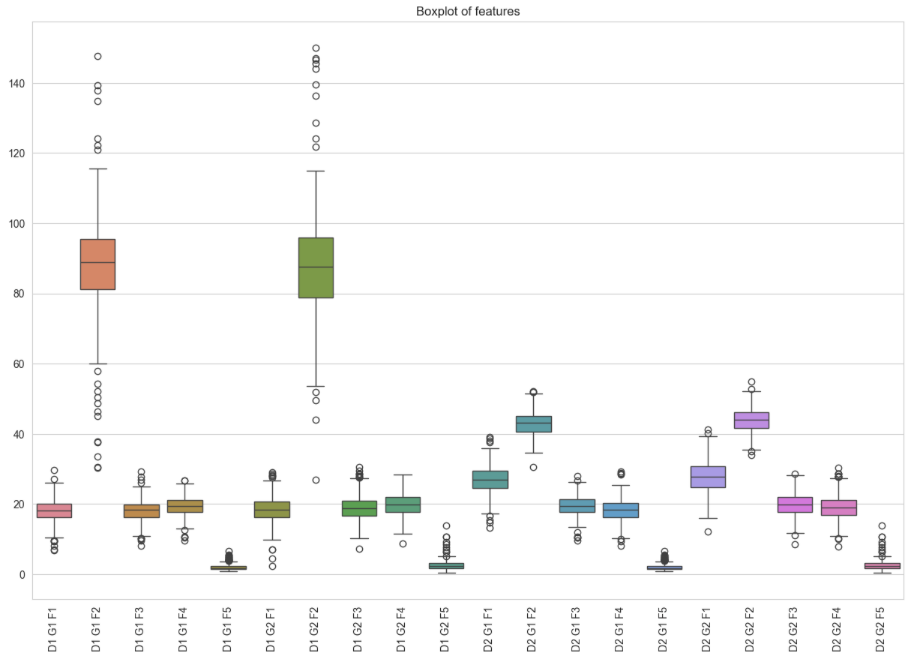


Figure : Boxplot for all features in the 2 datasets

The image shows various insights:

1. The range of F2 in the first dataset is significantly higher compared to all other features which again implies the need of some kind of normalization.
2. There are multiple outliers which could be useful in feature engineering to introduce new features related to outliers, but since the data sets are very small, outliers might not be very useful.

The last assessment was to draw a correlation matrix to see in a numeric readable way how much these features are correlated to the output:

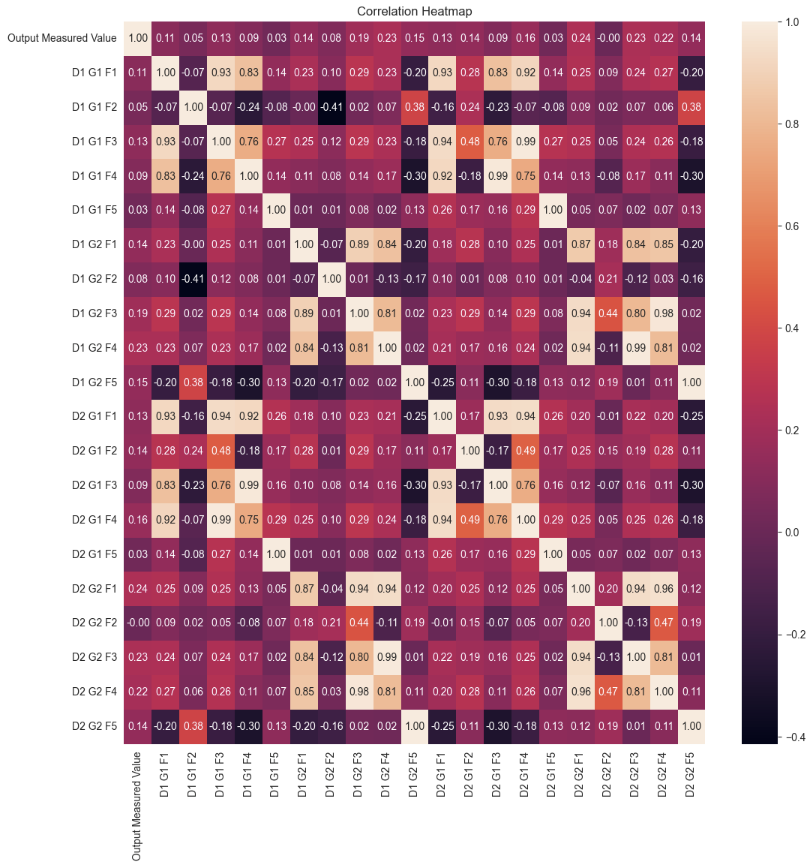


Figure : Correlation matrix between features

The correlation matrix shows a low correlation between all the features and the Output value. It also shows a high correlation between F1, F3 and F4 which.

It also shows high correlation between the same feature in different data sets.

## 2.4. Preparing multiple data frames for experimenting

The first step in data preparing was to normalize all values in all features as the initial assessment suggests unifying the scale and the shape of the data to avoid any possible issues while experimenting models.

Since the original data frame contains 2 different datasets in a form of spatial data. It was crucial to make various splits to make it ready for different explorations and experimenting. And the following data frames has been introduced initially:

1. ‘df1’ and ‘df2’ where each data frame represents a different data set from the 2 data sets available.
2. ‘df1g1’, ‘df1g2’, ‘df2g1’, and ‘df2g2’ where each data frame represents one side of the buildings for each data set.
3. ‘flattened\_df1’ and ‘flattened\_df1’ to convert the data into a simple 36 rows suitable for model trainings. These data frames were created by changing the feature format to include position, for example, ‘D1 G1 F1 P1’ which will represent F1 in position 1 in the first group of a building in the first data set.
4. ‘flattened\_df1g1’ and ‘flattened\_df1g1’

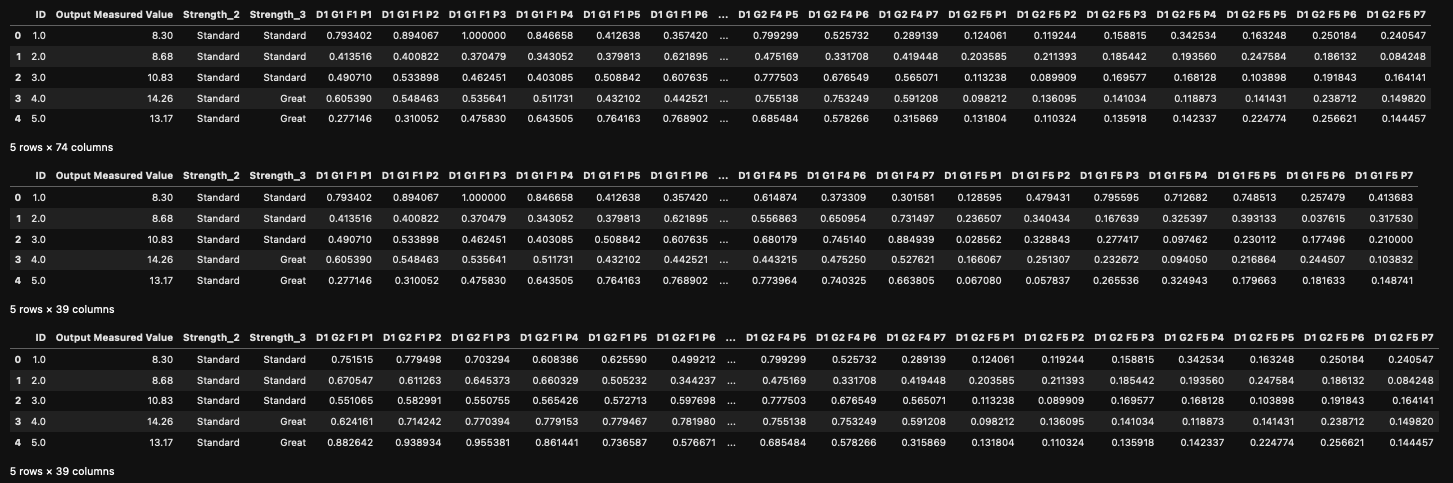


Figure : Flattened normalized data

# 3. Data analysis

## 3.1. Position analysis

### 3.1.1. Mean and Std aggregation.

A simple way to tackle the spatial data instead of flattening it, is aggregating the features on the positions, by calculating the mean, standard deviation for every feature in every building across all positions.

A new data frame with 36 rows was created with 40 features (10 x 2 (data sets) x 2 (mean and std) for each feature). Then the correlation was calculated between these new features and the Output.

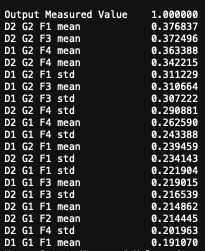


Figure : Correlation between aggregated features and the Output

The image above shows a better correlation for the aggregated features compared to each feature alone which suggest a relationship between different positions that impacts the overall output, and such relationship should be considered in model training.

### 3.1.2. Position differences, cumulative summation and multiplication.

Instead of extracting relationships that describe all positions together we could focus on relationships between the adjacent positions, like finding the difference in feature values between each 2 positions (P2 – P1, P3 – P2 … P7 – P6). We could also find the cumulative summation and multiplication for each position which could highlight some hidden insights.

For this purpose, a new data frame was created ‘df\_with\_diff\_cum’ and the correlation between new features and the Output was as follows:

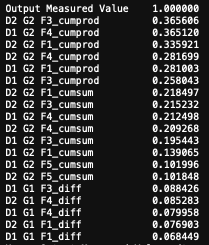


Figure : Correlation between adjacent positions and output

The result shows a similar correlation to the mean and std aggregation with better correlation on the cumulative operations which also suggests the spatial data relationship with output.

### 3.1.3. Distribution of Features by Position.

To better understand the ranges of each feature across all position, a boxplot for each feature in each position was plotted as following:

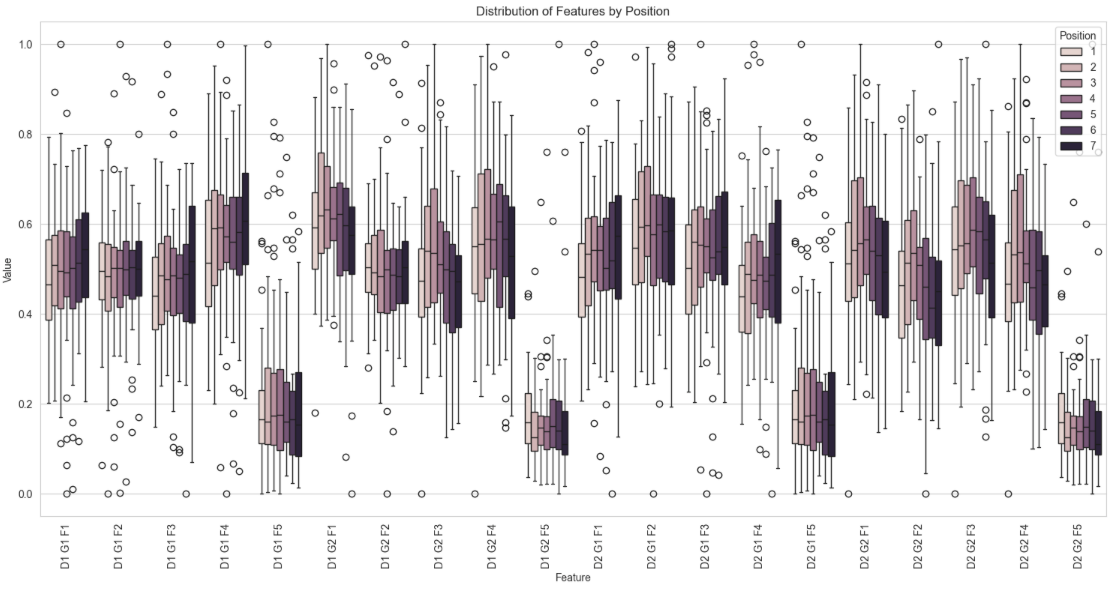


Figure : Boxplot of features per positions

The results show a similar distribution of data across all positions for each feature with small differences.

It also shows that the upper bound of values for the first group in the 7th position is higher than other features. Other than that, it shows that feature 5 most values are less than 0.3 with many outliers which might indicate some importance of these outliers for this specific feature in determining the Output.

### 3.1.4. Analyze similar buildings.

Since there are no strong relationships found, and since the data suggests a complex non-linear relationship between different features and positions, taking a sample of buildings with similar output values and analyze it could be useful. This approach could help us finding some kind of patterns that we could build on.

The image below shows a line chart with the values of each feature across each position for each one of the 3 building with similar output (~6):

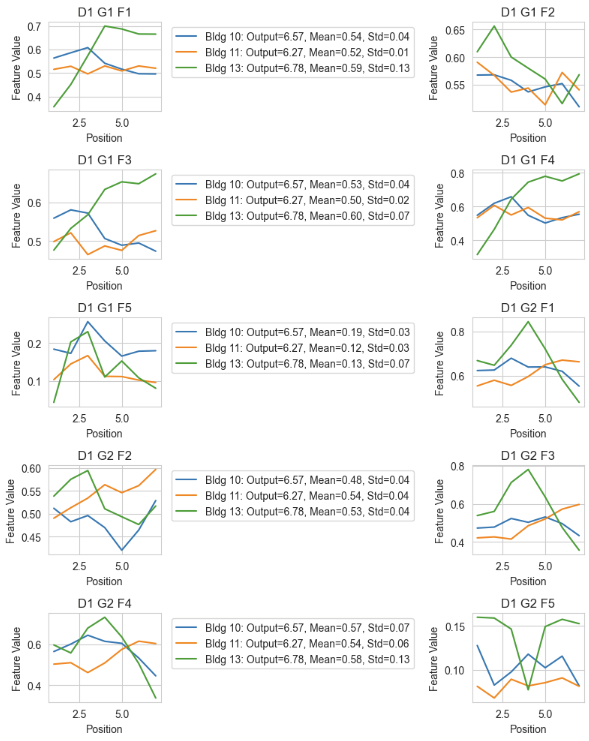


Figure : Similar building analysis

The lines look random and looking at line charts alone doesn’t provide useful information. On the other hand, looking at the mean of these 3 buildings shows a similar value which indicates that the mean can play a good role in classifying or predicting the output.

## 3.2. Flattened data Analysis.

### 3.2.1. Flattened features analysis.

Until now, we didn’t try to find relationships between flatten features with a format that includes the position ( . The results of finding the correlation between the flattened data and the Output was as following:

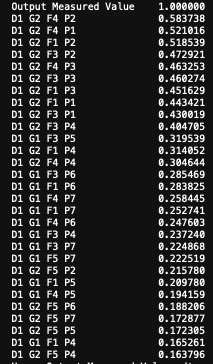


Figure : Flattened feature v.s. Output

The results show the best correlation so far between any feature and the output with multiple values above 0.4. This suggests focusing on handling the data in a flatten way.

### 3.2.2. Flattened features product analysis.

Since the flattened features showed some improvements in the correlation with the output, it might be beneficial trying to extract new features from the flattened ones by multiplying them with each other. The results of finding the correlation between the multiplied flattened data and the Output was as following:

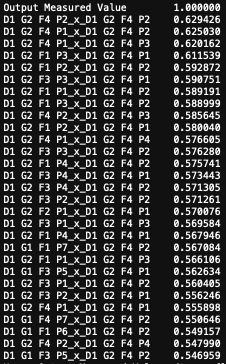


Figure : Multiplication of flattened features

A significant improvement in the correlation were observed with ~30 features with a correlation higher than 0.55. Transforming the flattened data to this form could have potential in fitting a good model.

## 3.3. Clustering analysis.

Clustering could be useful in group the data to groups with something in common which could reveal some hidden patterns, and which could be useful in features engineering by extracting features based on clusters.

I tried to cluster the data into 2 clusters as shown in the image below:

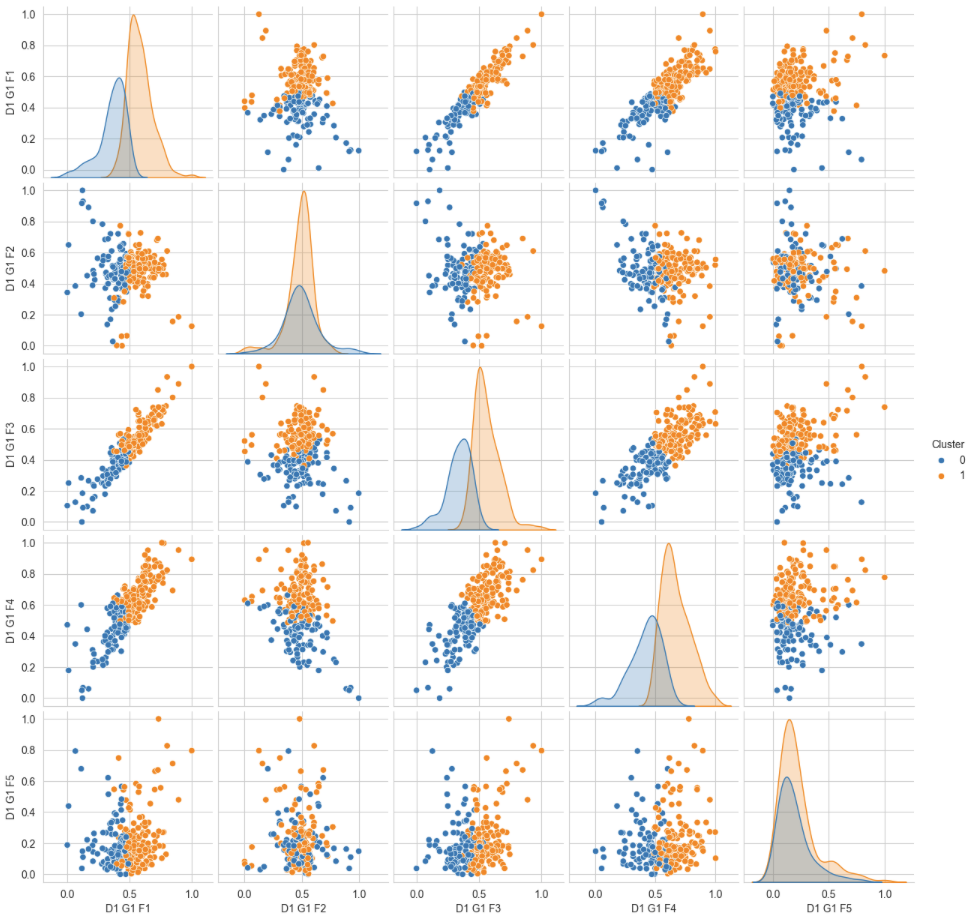


Figure : Clustering analysis for D1G1

The results shows that the data KMeans was able to split F1, F3, F4 and F5 into 2 clusters with similar number of elements in each cluster which suggests that we could use features such as cluster index and distance to cluster centroid as feature in the model training.

This also suggests that we should try training the model with and without F2 to see if it has negative impact since it consistently showed some randomness.

# 4. Model training experiments

## 4.1. Training Methodology

The data sets provided are very small which represents a set of features for only 36 buildings. The classic training and testing splits techniques could be problematic since a range of 10-30% test data sets only means 2-10 samples which will eventually lead to a not reliable results especially for classification experiments and it won’t cover various sample from various strengths. So, instead of that, a cross-validation was used in all experiments with 5 folds to ensure the model generalize well on all subsets of data and to ensure the results are reliable as much as possible.

Since the data analysis showed what looks like a complex non-linear relationship, doing a regression a lone might not work. That’s why 2 new columns were introduced to categorize outputs into binary classification and multi-class classification with the categories.

The experiments used 3 different models for experiments:

1. Linear regression model: It was decided to start with a straightforward approach with a basic model that will try to fit the features linearly which give us a starting point for more exploration.
2. Random forest regressor: Random Forest knows to its good ability of generalizing model to avoid overfitting and the randomness in building its subtrees and it’s split could be beneficial in finding non-linear relationship. [1]
3. Random forest classifier: For the same reasons as random forest regression, the random forest classifier will be mainly used to classify the data into 2 or 3 categories. Another main reason for choosing random forest was based on a personal preference since I have most experience with this model. [1]

To get the most out of each model, a hyperparameters tunning was conducting using hundreds of parameters options trying to boost the accuracy as much as possible. The small size of the data and the high computational resources available made it viable to explore with as many options as possible.

The evaluation metrics used where RMSE (Root mean square error), for regression and average accuracy a cross all folds for classification in addition to ROC and AUC scores.

PCA will also be used trying to reduce data dimensionality.

## 4.2. Experimentation

### 4.2.1. Linear regression and Random Forest for flattened data and for aggregated data

The first few experiments targeted the aggregated positions data, the flattened data, and the flattened product data.

The experiments targeted both datasets using linear regression, random forest regression and random forest classifier.

|  |  |  |  |
| --- | --- | --- | --- |
| Data used | Model type | Best parameters | Accuracy |
| Aggregated df1 | Linear regression | - | RMSE = 9.67  = -3.86 |
| RF regressor | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_split': 10, 'n\_estimators': 200} | RMSE = 4.78  = - 0.18 |
| RF binary classifier | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100} | 64.3% |
| RF classifier (3 categories) | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 50} | 50% |
| Aggregated df2 | Linear regression | - | RMSE = 12.26  = -6.81 |
| RF regressor | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_split': 10, 'n\_estimators': 50} | RMSE = 4.79  = - 0.195 |
| RF binary classifier | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 50} | 69.3% |
| RF classifier 3 categories) | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 50} | 49.6% |
| Flattened df1 | Linear regression | - | RMSE = 10.47  = -4.69 |
| RF regressor | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_split': 10, 'n\_estimators': 50} | RMSE = 4.41  = - 0.011 |
| RF binary classifier | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 100} | 74.6% |
| RF classifier (3 categories) | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 2, 'n\_estimators': 200} | 63.9% |
| Flattened df2 | Linear regression | - | RMSE = 14.38  = - 9.74 |
| RF regressor | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_split': 10, 'n\_estimators': 50} | RMSE = 4.38  = 0.002 |
| RF binary classifier | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 2, 'min\_samples\_split': 5, 'n\_estimators': 50} | 68.9% |
| RF classifier (3 categories) | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 50} | 69.3% |
| Flattened product df1 | Linear regression | - | RMSE = 14.34  = -9.69 |
| RF regressor | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_split': 5, 'n\_estimators': 200} | RMSE = 4.2  = 0.08 |
| RF binary classifier | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 50} | 77.5% |
| RF classifier (3 categories) | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 50} | 72.1% |

Figure : Linear regression and Random Forest for flattened data and for aggregated data

The linear regression showed a very bad results with a very high RMSE ranges ~10-14 and negative R-suqared for all data frames used. These RMSE values with the original output ranges between 2- 18 suggests a high failure in fitting the data in this linear model.

On the other hand random forest regressor showed a significatly better results acrossed different dataframes with RMSE around 4 and R-squared around zero. Even though, its way better than linear regression, it’s still far away from a resonable error rate for such small range of outputs.

Finally the binary and multi-class classifers showed a good start with a highest accuracy of 72.1% for multi-class and 77.5% for binary class using the flattened product dataframe.

The classification showed the worst results for the the aggregated positions data compated to the flattened positions which also suggests to focus the upcoming expirments on the flattened data.

Both data sets showed similar results which make it reasonable to focus on one dataset for the upcoming expirments then after obtaining the desired results, the same approach can be applied to the other data set to reach conclusions.

### 4.2.2. Flattened data with outliers-related features.

For this experiment the flattened dataset 1 were used and an extra 4 features were added:

1. Number of upper-bound outliers for each building.
2. Number of lower-bound outliers for each building.
3. Whether the building has upper-bound outliers.
4. Whether the building has lower-bound outliers.

|  |  |  |  |
| --- | --- | --- | --- |
| Data used | Model type | Best parameters | Accuracy |
| Flattened df1 with extra outliers related features | RF regressor | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_split': 10, 'n\_estimators': 200} | RMSE = 4.38  = 0 |
| RF binary classifier | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 4, 'min\_samples\_split': 2, 'n\_estimators': 100} | 71.8% |
| RF classifier (3 categories) | {'max\_depth': 10, 'max\_features': 'sqrt', 'min\_samples\_leaf': 1, 'min\_samples\_split': 2, 'n\_estimators': 50} | 69.3% |

The results didn’t show any improvements compared to the previous experiments, which might indicate that the measured outliers are not actual outliers, and the main reason they were identified as outliers is the small size of the data.

### 4.2.3. Trying all combinations of positions.

In this expirment, the data will be flattened and all the combinations of positions will be explored. This expirment aims to find which combination of positions could lead to better relationships between the features and the output.

The combinations includes: (P1)(P2)…(P7)(P1 P2)…(P6 P7) …. (P1 P2 P3 P4 P5 P6 P7). The number of combinations is 127 ().

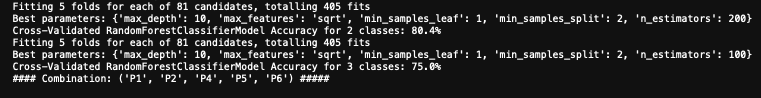


Figure : All positions combinations expirment

The best results for the 127 experiments were for positions ('P1', 'P2', 'P4', 'P5', 'P6') with multi-class accuracy of 75% and binary classification accuracy of 80.4% which are the best results obtained so far. This highlight the importance of the correlation between diffirent positions.

### 4.2.4. Trying clustering realted features

In this expirment, the flattened data will be clustered using different number of clusters, for each setting, the cluster index will be used as a feature, in addition to features distance from each cluster centroid.

The experiment ran on different number of clusters ranged from 2 to 12.

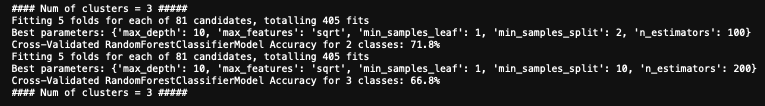


Figure : Clustering experiment

The best obtained result was using 3 clusters with accuracy of 72% for binary classification and an accuracy of 67% for 3 categories classification.

The clustering didn’t improve the results, but it could be beneficial if merged with other feature engineering techniques.

### 4.2.5. Trying PCA to reduce dimensionality.

The flattened data has a relatively large number of features compared to the number of samples. In this experiment, the dimensionality of flattened data was reduced using PCA with different number of PCA components (6, 12, 18, 24, 30, and 36). 36 is largest number of components that could be used because of the number of samples.

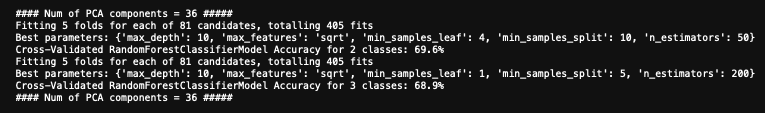


Figure : PCA experiment

The best results came from 36 components which the largest possible options to use. The results had lower accuracy compared to the flattened data results on its own. This indicates that PCA won’t be very helpful in this kind of data set because of the low number of samples which limits the number of components that could be used.

### 4.2.6. Splitting the output using weighted sum for features.

In this experiment, instead of flattening the data and getting only 36 samples, the output will be splitted between the 7 positions for each building using a weighted sum of features. Meaning that the position with the highest sum of features values will get the highest proportion of the output.

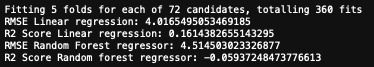


Figure : Splitting the output using weighted sum.

This experiment gave the best regression results so far with an RMSE = 4 and R-Squared – 0.16. But still the results are not great considering the range of output of values.

### 4.2.6. Using 14 positions with weight correlation with output.

In this experiment, the group 2 features will be transformed to form another 7 positions which will result in 14 positions per building, in addition to that, each feature in each position correlation with the output will decide how the output will be distributed over the 14 positions.

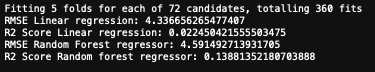


Figure : Using 14 positions with weight correlation with output.

The results showed similar values compared to the previous experiment with a slight advantage to the previous one.

The last 2 experiments showed the best linear regression results so far.

### 4.2.7. Using deep learning

Another approach used was creating a multi-layer neural network with many hidden layers that could potentially capture non-linear patterns in the data. For this purpose, 6 hidden layers were used with 16 neurons in each with Relu as an activation function.

This experiment tried both regression, binary, and multi-class classification using the same model architecture with evaluation based on 5 number of folds and averaging the results.

* Avg RMSE: 5.195316390218844
* Avg R2 Score: -0.6948699653837845
* Avg Binary Classification Accuracy: 64%
* Avg Multi-Class Classification Accuracy: 50.3%

The results didn’t show any improvments which suggests that the data were overfit on the training phase. And there are no poteniatl solution for this with this small number of data which makes deep learning a bad expirmentation choice.

### 4.2.8. Using rolling window

One of the great features provided by pandas is the rolling features, where you can take a slice of data (In this case, a slice of positions) and generate some aggregations on that window such as summation, mean, min, max … etc. [2]

In this experiment, 5 windows size were tested (2 to 6) with rolling on sum, mean, median, variance, and skewness.

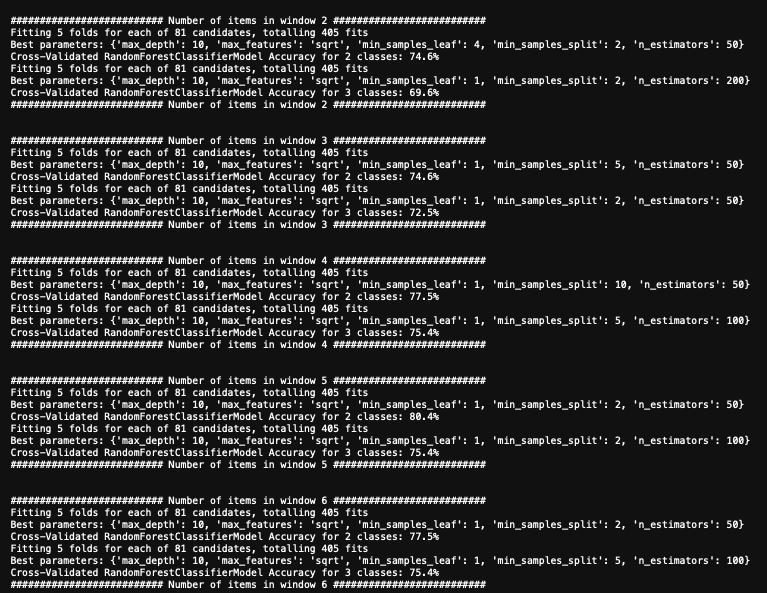


Figure : Rolling windows.

This approach gave a good result on number of windows = 5, with an accuracy of 77.5% for multi-class and 80.4% for binary classification. Merging this approach with other approaches could enhance the overall results.

### 4.2.9. Hybrid approach between clustering and rolling windows.

For the final experiment, a different combination of previous experiments was merged aiming to getting the best of each feature engineering techniques. The final experiment was a combination of clustering approach and rolling window.

The flattened data 1, flattened data 2, flattened data 1 group 1, flattened data 1 group 2, flattened data 2 group 1, flattened data 2 group 2 where each tested using multiple configurations including 8 number of clusters options, 6 window sizes, and 6 different PCA options with a seventh option without PCA, which give us 334 experiments for each data frame.

All data frames were tested using binary classification and 3-classes classification and the accuracy was calculated using cross-validation with 5 number of folds to ensure the reliability of the test results.

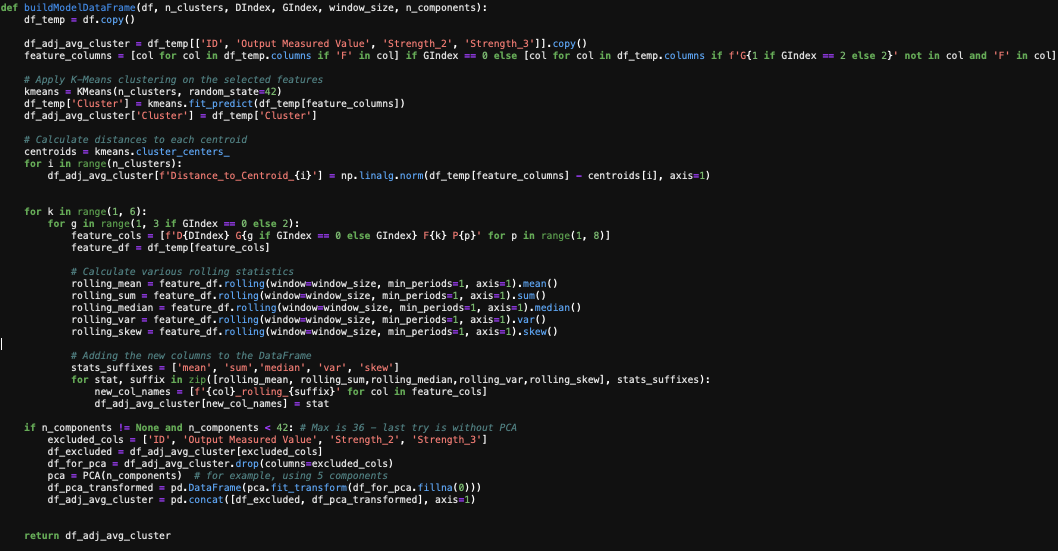


Figure : Code snippet from the final experiment

This experiment gave the best results with a cross-validation accuracy of 89% for binary classification and a cross-validation accuracy of 81% for multi-class classification.

# 5. Results analysis

The last experiment gave the best accuracy for the flattened dataset 1 and the second-best accuracy was from the second group in the second dataset.

The best results for all used data frames can be summarized as the following:

|  |  |  |  |
| --- | --- | --- | --- |
| Data set | Classification | Configuration (PCA wasn’t use in any) | Accuracy |
| Flattened\_df1 | Binary | Number of clusters = 6, Number of window items = 4 | 89% |
| Three categories | Number of clusters = 5, Number of window items = 4 | 80.7% |
| Flattened\_df1g1 | Binary | Number of clusters = 3, Number of window items = 4 | 69% |
| Three categories | Number of clusters = 8, Number of window items = 4 | 64% |
| Flattened\_df1g2 | Binary | Number of clusters = 4, Number of window item = 7 | 86.1% |
| Three categories | Number of clusters = 4, Number of window items = 3 | 78% |
| Flattened\_df2 | Binary | Number of clusters = 3, Number of window items = 6 | 86.1% |
| Three categories | Number of clusters = 4, Number of window items = 3 | 75% |
| Flattened\_df2g1 | Binary | Number of clusters = 6, Number of window items = 4 | 66.4% |
| Three categories | Number of clusters = 5, Number of window items = 2 | 53% |
| Flattened\_df2g2 | Binary | Number of clusters = 11, Number of window items = 5 | 89% |
| Three categories | Number of clusters = 6, Number of window items = 3 | 78% |

Figure : Hybrid approach results

## 5.1. Flattened data set 1 analysis

|  |  |
| --- | --- |
| Binary classifcation 89% accuracy | Multi-class classification 80.7% accuracy |
|  |  |

1. For binary classifaction: The results shows an 89% overall accuracy with 1 out of 23 as false positives and 3 out of 13 as false negatives.
2. For multi-class: The resutls shows an 80.7% overall accuracy with 2 out 13 was not predicted correctly for “weak class” which give us a better result compared to the binary classifciation if the main goal is to always detect the weak buildings.
3. For the multi class the “great” category gave the worst results with only 3 out of 7 were correctly identified.

## 5.2. Flattened data set 2 group 2 analysis

|  |  |
| --- | --- |
| Binary classifcation 89% accuracy | Multi-class classification 78% accuracy |
|  |  |

1. For binary classifaction: The results are identical to the flattened data set 1
2. For multi-class: The resutls shows an 78% overall accuracy with 3 out 13 was not predicted correctly for “weak class” which give us the same result compared to binary which suggests binary classifier if we decided to use this data frame is better.

## 5.3. Recommendations

If identifying the weak building is top priortiy, then using all the values in dataset 1 in it’s flattened form is recommended since it resulted in the best accuracy to identify weak buildings.

If the overall accuracy is the most important, then using the binary classifier in the dataset 2 group 2 is the best option since it will give the highest accuracy and it will get rid off the need of taking mesaure from one side of the building.

There is a room for deeper analysis and model improvements if we gain more information about the data from the data owner and a domain expert. The conducted analysis was mostly blind with multiple expirements hoping for a good fitting for data.

# 6. Conclusion

In the report, we analyzed spatial data using various ways trying to find patterns and relationships between the different features, this included using different kind of plots like histograms, scatter plots, box plots and correlation matrix heatmap. We also restructered the original data using multiple ways like flattening it, aggregating it and taking different subsets.

Multiple expirements has been conducted based on different feature engineering techinques to find the hidden relationships between the features and the positions. Each expirment conducted on different models, different parameters, and different forms of data. We explored some advanced techinques like rolling window which helped in improving the overall model performance. We ended up with a hybrid approach that combine two techniques together which are clustering and rolling window aggregations to build the best model.

The analysis and expirments showed a poor performance of regression which lead us to focus on classification which resutled in a good accuracy of 89% usign binary classification and 80.7% using three categories classification. Building the model using the first dataset in it’s flatten form gave the best results, and we got a similar results using only features from the second side of the building of the second data set. The multiclass classfication gave the best results for separting the weak category from other with an accuracy of 85.4% (One vs All).

It’s crucial to get more information about the data for a future improvments to achieve a more targeted analysis that could eventually lead into a strong fitting of data.

7. References

|  |  |
| --- | --- |
| [1] | G. L. Team, "Random forest Algorithm in Machine learning: An Overview," 13 6 2023. [Online]. Available: https://www.mygreatlearning.com/blog/random-forest-algorithm. |
| [2] | "Introduction to Pandas rolling," Educba, 7 6 2023. [Online]. Available: https://www.educba.com/pandas-rolling/. |