# TITLE OF THE REPORT

Computer Aided Solution Processing, 2021

Menna-Allah Mohamed, 210ADB005

# 1. Introduction

Regression models are used to predict values that depend on some factors by computing previous data and building assumptions. They are very useful when the output is correlated to the inputs. In this project, two data sets will be discovered and used to apply several regression methods. The first data set considers the heating load of building as the output that depends on the building parameters. On the other hand, the second data set considers the concrete strength as the output that depends on its age and amounts of ingredients. In both data set, three regression models will be applied which are linear regression, k-nearest neighbors regression and random forest regression.

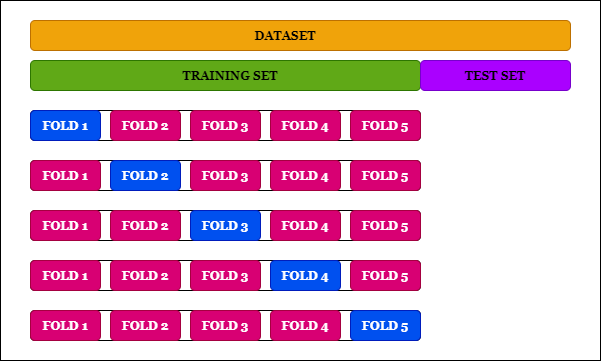
# 2. Literature Review

In this section, there will be short explanation about k-fold cross validation method, the regression models that will be used in this project, and how it will be tested.

## 2.1. K-Fold Cross Validation

Using dataset to predict futuristic values requires dividing it into training set to produce the model, and test set to calculate the accuracy of the model. Sometimes, there are many models to choose between them, so we take a part from the training set which is called validation set to compare and choose between those models. To provide an efficient comparison, k-fold cross validation method is used.

K-fold validation method divides the training set into k segments. It performs k-iterations, in the iteration it uses the segment to be the validation set, while other segments become the training set. The accuracy of that model is calculated by the average of the k-scores from each iteration. Figure 1 shows the division of the data set while applying 5-fold cross validation method.



**Fig. 1.** K-fold cross validation method [1]

## 2.2. Multiple Linear Regression

Multiple linear regression is a statistical method that is used to predict an output that respond to other parameters. It models the relationship between the output and other inputs linearly as each input has its factor effect . This relation could be driven as the following:

(1)

where *Y* is the response variable, is the model error,

is the input (explanatory variable) where and is the number of inputs,

is the y-intercept (constant) and is the factor where and is the number of inputs.

As any regression model, multiple linear regression uses previous data to design and test the model. In the designing process, the explanatory variables and the response variable are given, so the aim is to find the factors . By many mathematical proofs, we can get those factors by the following formula:

(2)

where *Y* is a matrix of the response variable, is a matrix of the explanatory variables and is the matrix transpose of the explanatory variables in the training data.

## 2.3. K-Nearest Neighbors (K-NN) regression

The K-Nearest Neighbors is a method that is used for regression and classification. As both data sets have numerical response output, the k-nearest neighbors method is used here for regression by returning the average of the nearest neighbors. In this method, the training process is only storing the records. In the prediction method, the nearest neighbors are detected as the k-points which have the least Euclidian distance that follow the equation below:

(3)

where is the query point, is the point that we need to calculate the distance with it, is the feature, and is number of features.

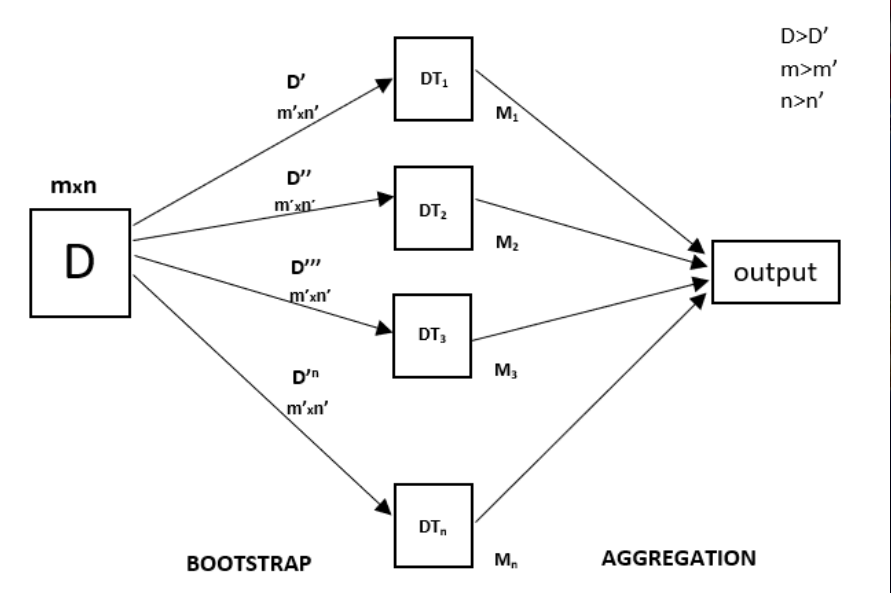
After selecting the k-neighbors, the predicted value could be calculated as:

(4)

where is the weight of the point (if uniform weight is used, it will be equal 1, else it will be the inverted distance).

## 2.4. Random Forest regression

The random forest is a regression method that is composed of parallel independent decision trees. This technique overcomes the drawback of the high variance of the decision trees, because it considers all the results produced from each decision trees. Figure 2 illustrates the division of the whole data into random rows (records) with some columns (features) to go through a decision tree; the result is the average of those decision trees’ results.



**Fig. 2.** Random Forest Regression [2]

## 2.5. Regression Model Testing

In order to test any model, there are many methods that can be used. This project will consider two methods: R-squared method (shown in eq. 5), and Mean Squared Error (shown in eq. 6)

(5)

(6)

where is the response variable of the record, is the predicted response variable of the record, is the mean of the response variable, and n is the number of the tested records.

# 3. Overview of the used data

## 3.1. Dataset 1 (Energy Efficiency Data Set)

By discovering the first dataset, it had been found that: the dataset size is 6912, number of features (columns) is 8 features (shown in Table 1), number of records (rows) is 768 records, number of null values is zero, and the response variable is Y (Heating Load).

**Table 1**

First Dataset Features

|  |  |  |
| --- | --- | --- |
| Feature number | Feature name used in the dataset | Feature meaning |
| 1 | X1 | Relative Compactness |
| 2 | X2 | Surface Area |
| 3 | X3 | Wall Area |
| 4 | X4 | Roof Area |
| 5 | X5 | Overall Height |
| 6 | X6 | Orientation |
| 7 | X7 | Glazing Area |
| 8 | X8 | Glazing Area Distribution |

## 3.2. Dataset 2 (Concrete Compressive Strength Data Set)

By discovering the second dataset, it had been found that: the data size is 9270, number of features (columns) is 8 features (shown in Table 2), number of records (rows) is 1030 records, number of null values is zero, and the response variable is Y (Concrete compressive strength (MPa, megapascals)).

**Table 2**

Second Dataset Features

|  |  |  |
| --- | --- | --- |
| Feature number | Feature name used in the dataset | Feature meaning |
| 1 | X1 | Cement (component 1) (kg in a m^3 mixture) |
| 2 | X2 | Blast Furnace Slag (component 2) (kg in a m^3 mixture) |
| 3 | X3 | Fly Ash (component 3) (kg in a m^3 mixture) |
| 4 | X4 | Water (component 4) (kg in a m^3 mixture) |
| 5 | X5 | Superplasticizer (component 5) (kg in a m^3 mixture) |
| 6 | X6 | Coarse Aggregate (component 6) (kg in a m^3 mixture) |
| 7 | X7 | Fine Aggregate (component 7) (kg in a m^3 mixture) |
| 8 | X8 | Age (day) |

# 4. Experiments

Each dataset is divided into two parts, train set (80% of the dataset), and test set (20% of the dataset). Then, a model had been prepared for cross validation with k equals 10. Now, the scores of the cross validation could be obtained by selection the regression method. In order to choose the best model, three parameters will be considered in evaluation each regression model which are the following:

1. Implementation Time: the time taken to set the best parameter and end up with the final model.
2. Fitting Time: the time taken by each model to produce the cross-validation scores.
3. Performance: the cross-validation scores average (they were evaluated by R-squared method)

# 

## 4.1. Experiments with the first dataset

### *4.1.1 Linear Regression Model*

As explained in the literature section, the linear regression is simple to be obtained by performing some computations on matrices. That’s why the built-in linear regression function has a few parameters that did not affect the evaluating parameters of the regression model. The final model results the following:

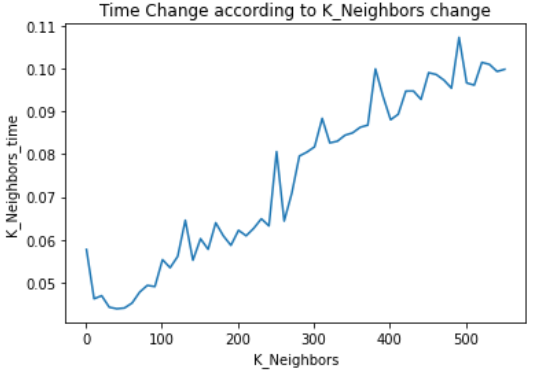
1. Implementation Time = 0.04963148400020145
2. Fitting Time = 0.033060394000131055
3. Performance = 0.9177663224666542

### *4.1.2 K-Nearest Neighbors Regression Model*

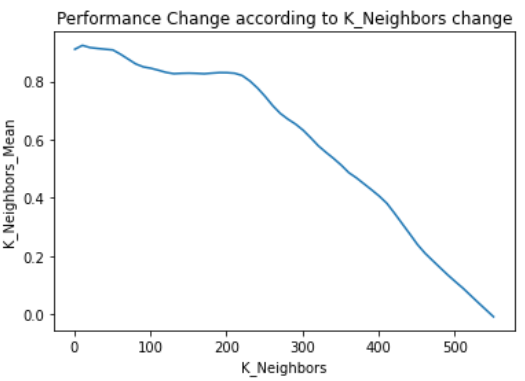
In the K-Nearest Neighbor regression, there are two parameters that should be considered:

1. The number of chosen neighbors (k):

While the number of used records equals to 90% of the training set, it had been found that number of chosen neighbors could vary from 1 to 522. In order to choose the best one without spending too much time, a for loop was used with a step equals 10. Figure 3 shows the time spent to evaluate each k, and figure 4 shows the performance vs k. As the number of chosen neighbors increases, the needed time increases. The performance showed its peak when the k equals 11.



**Fig. 3.** Time Change According to K Change



**Fig. 4.** Performance Change According to K Change

1. The weight calculation criteria:

If the weight of each neighbor is uniform, it means that all neighbors will have the same effect on the predicted response. On the other hand, choosing the weight as “distance” will put the Euclidean distance parameter as a factor in predicting the response variable, so the nearer points will have more effect on the response. During the comparison between the best performances from either uniform or distance, the results (shown in table 3) showed that the time difference were neglectable, but the distance weight had the best performance.

**Table 3**

Weight Criteria Results

|  |  |  |
| --- | --- | --- |
|  | Uniform Weight | Inverted Distance Weight |
| Fitting Time | 0.054953296999883605 | 0.04811888599942904 |
| Performance | 0.9235221657202514 | 0.9139946074096349 |

After choosing the best k which was 11 and the best weight calculation criteria which was distance, the final model results were as the following:

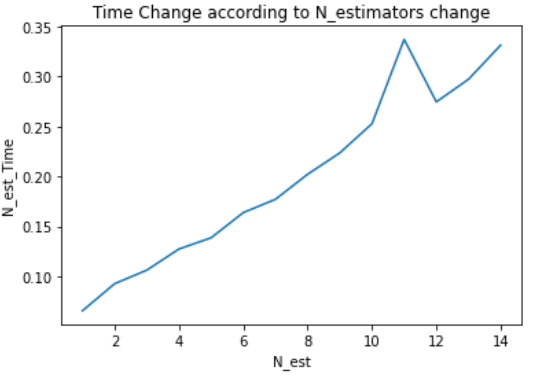
1. Implementation Time = 4.792722084999696
2. Fitting Time = 0.04946877800011862
3. Performance = 0.9235221657202514

### *4.1.3 Random Forest Regression Model*

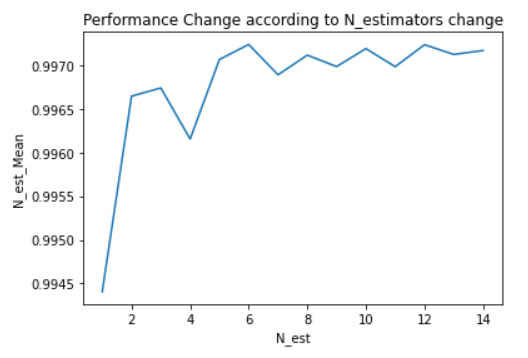
In the Random Forest regression, there are many parameters that could be changed in the built-in function; however, some parameters could be ignored as they do not have a significant effect on the time. Here, we will consider only two parameters:

1. The number of decision trees (n estimators):

The default value of this parameter is 100 which takes too much time compared to other regression methods. Thus, choosing the number of decision trees were important to reduce this time without having a big influence on the performance. The number of decision trees varied by using for loop from 1 to 15 in which the time and the performance in each iteration were saved. Figure 3 shows the time in each iteration, while figure 4 shows the performance in each iteration.



**Fig. 5.** Time Change According to N-estimators Change



**Fig. 6.** Performance Change according to N-estimators Change

By observing the previous figures, it could be noticed that the time increases almost linearly while the performance got a steady state after having 5 decision trees. Therefore, the number of decision trees could be chosen as 5 decision trees.

1. Error Evaluation Criteria:

Evaluation the quality of a split in each decision tree could be done by several evaluation methods. In the used functions, the mean squared error (MSE) and the mean absolute error (MAE) were supported. In table 4, the results of using each criterion showed that the performance have no significant difference, while the time has high difference as the MAE time is more than double the MSE time.

**Table 4**

Error Evaluation Criteria Results

|  |  |  |
| --- | --- | --- |
|  | MAE | MSE |
| Fitting Time | 0.4188100800001848 | 0.14016991200060147 |
| Performance | 0.9967953079375184 | 0.9967775913159196 |

After choosing the best number of decision trees which was 5 and the best error evaluation criteria which was MSE, the final model results were as the following:

1. Implementation Time = 3.880314568999893
2. Fitting Time = 0.14348316700034047
3. Performance = 0.9970218586061323

### *4.1.4 Choosing Best Model*

As mentioned before, choosing the best model could be done by considering three parameters; implementation time, fitting time, and performance. As shown in table 5, each model had its own best record if we compare them upon a specific parameter. Considering the implementation time, the linear regression model has the best implementation time as it has no hyperparameters to be chosen. On the other hand, the K-Nearest Neighbors regression was recorded as the best fitting time model as it depends on the number of records which was almost small. Regarding performance analysis, the random forest had the best performance result as it depends upon the results of many decision trees.

**Table 5**

Results of Regression Models

|  |  |  |  |
| --- | --- | --- | --- |
| Models | Linear Regression | K-NN Regression | Random Forest Regression |
| Implementation time | 0.049631 | 4.792722 | 3.880315 |
| Fitting Time | 0.048469 | 0.047568 | 0.143483 |
| Performance | 0.917766 | 0.923522 | 0.997022 |

### *4.1.5 Testing Best Model*

Testing the best model was done by using the test set and compare the predicted values with the original values. The comparison was done using mean squared error (MSE) method. As mentioned before, we could not call a model as the best model in all parameters. Thus, each model of them could be the best model if we consider different parameter in the comparison (All of them were used to be the best model).

By fitting the whole training set into the chosen model, the model became ready for the prediction process. Finally, it could be used with the testing set to predict the response variable. The accuracy of the prediction technique was evaluated by comparing the predicted results with the original results using MSE method. The results of the models were shown in table 6

**Table 6**

Accuracy of the best Models

|  |  |  |  |
| --- | --- | --- | --- |
| Best Model | Linear Regression | K-NN Regression | Random Forest Regression |
| MSE | 9.936294219141887 | 8.719853939036172 | 0.18622877922077888 |

## 4.2. Experiments with the second dataset

### *4.2.1 Linear Regression Model*

The second dataset had the same methodology as the first dataset, where there were no important hyperparameters to be considered. Thus, the final model results were as the following:

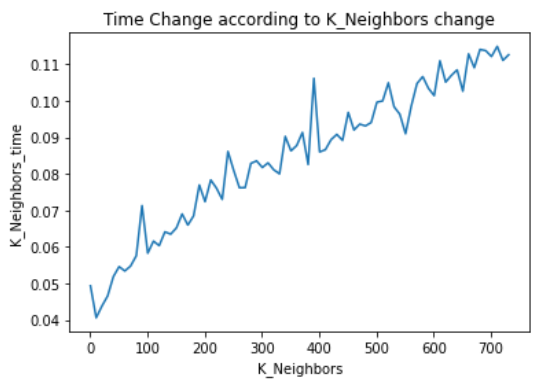
1. Implementation Time = 0.050153586999954314
2. Fitting Time = 0.036095572000022
3. Performance = 0.6255430339646376

### *4.2.2 K-Nearest Neighbors Regression Model*

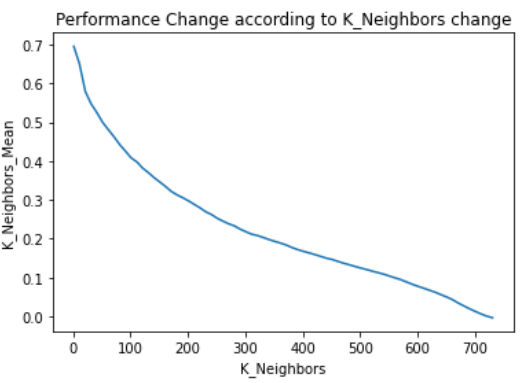
In the K-Nearest Neighbor regression, there are two parameters that should be considered:

1. The number of chosen neighbors (k):

While the number of used records equals to 90% of the training set, it had been found that number of chosen neighbors could vary from 1 to 741. In order to choose the best one without spending too much time, a for loop was used with a step equals 10. Figure 3 shows the time spent to evaluate each k, and figure 4 shows the performance vs k. As the number of chosen neighbors increases, the needed time increases. The performance showed its peak when the k equals 1.



**Fig. 7.** Time Change according to K change



**Fig. 8.** Performance Change according to K change

1. The weight calculation criteria:

During the comparison between the best performances from either uniform or distance, the results (shown in table 3) showed that there is no difference in the performance as it is only one point to be weighted, but the fitting time of the distance method is quite better, so it will be chosen.

**Table 3**

Weight Criteria Results

|  |  |  |
| --- | --- | --- |
|  | Uniform Weight | Inverted Distance Weight |
| Fitting Time | 0.11422154800004591 | 0.04086025399999471 |
| Performance | 0.694902556160428 | 0.694902556160428 |

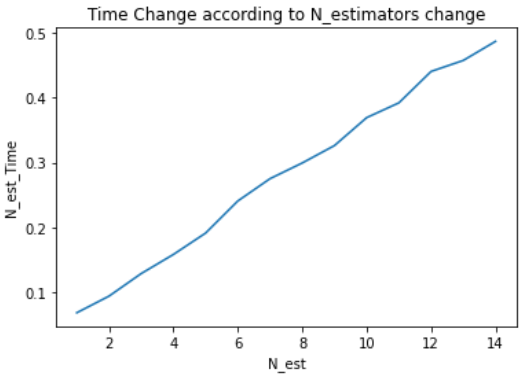
After choosing the best k which was 11 and the best weight calculation criteria which was distance, the final model results were as the following:

1. Implementation Time = 6.8640734079999675
2. Fitting Time = 0.042333943000016916
3. Performance = 0.694902556160428

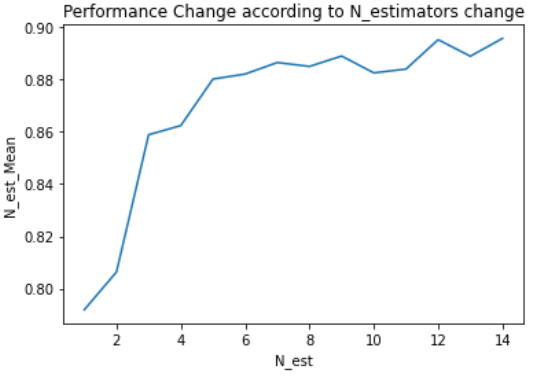
### *4.2.3 Random Forest Regression Model*

1. The number of decision trees (n estimators):

The number of decision trees varied by using for loop from 1 to 15 in which the time and the performance in each iteration were saved. Figure 3 shows the time in each iteration, while figure 4 shows the performance in each iteration. By observing the figures, it could be noticed that the time increases almost linearly while the performance got a steady state after having 5 decision trees. Therefore, the number of decision trees could be chosen as 5 decision trees.



**Fig. 5.** Time Change According to N-estimators Change



**Fig. 6.** Performance Change according to N-estimators Change

1. Error Evaluation Criteria:

Evaluation the quality of a split in each decision tree could be done by several evaluation methods. In the used functions, the mean squared error (MSE) and the mean absolute error (MAE) were supported. In table 4, the results of using each criterion showed that the performance have no significant difference, while the time has high difference as the MAE time is more than double the MSE time.

**Table 4**

Error Evaluation Criteria Results

|  |  |  |
| --- | --- | --- |
|  | MAE | MSE |
| Fitting Time | 1.0573958500000344 | 0.19881571500002337 |
| Performance | 0.8812129896244313 | 0.8754599278764796 |

After choosing the best number of decision trees which was 5 and the best error evaluation criteria which was MSE, the final model results were as the following:

1. Implementation Time = 5.735446935000027
2. Fitting Time = 0.20372850699993705
3. Performance = 0.8593986286045044

### *4.2.4 Choosing Best Model*

As mentioned before, choosing the best model could be done by considering three parameters; implementation time, fitting time, and performance. As shown in table 5, each model had its own best record if we compare them upon a specific parameter. Considering the implementation time, the linear regression model has the best implementation time as it has no hyperparameters to be chosen. It was also recorded as the best fitting time model because that the K-Nearest Neighbors regression depends on the number of records which was almost large and the random forest regression needs to run several decision trees in the fitting process. Regarding performance analysis, the random forest had the best performance result as its result was obtained as the average of many decision trees outputs.

**Table 5**

Results of Regression Models

|  |  |  |  |
| --- | --- | --- | --- |
| Parameters | Linear Regression | K-NN Regression | Random Forest Regression |
| Implementation time | 0.047005 | 7.001419 | 5.780271 |
| Fitting Time | 0.032884 | 0.038389 | 0.199034 |
| Performance | 0.625543 | 0.694903 | 0.879999 |

### *4.2.5 Testing Best Model*

The linear regression model was chosen as the best model when we compare according to the implementation time or fitting time (speed). According to the performance analysis, the random forest was the best. Thus, both were used to predict the response variable.

By fitting the whole training set into the chosen model, the model became ready for the prediction process. Finally, it could be used with the testing set to predict the response variable. The accuracy of the prediction technique was evaluated by comparing the predicted results with the original results using MSE method. The results of the models were shown in table 6. It could be noticed that although the random forest performance was better than the K-NN, the MSE in the testing set shows that the K-NN was better than the random forest regression.

**Table 6**

Accuracy of the best Models

|  |  |  |  |
| --- | --- | --- | --- |
| Best Model | Linear Regression | K-NN Regression | Random Forest Regression |
| MSE | 113.79979243508389 | 0.039606913656328235 | 20.09485087479116 |

## 4.3. Comparison of the results of both datasets

Applying the same methodology in two different datasets does not guarantee having the same output. Here, the similarities and differences between the results of both datasets will be discovered

### *4.3.1 Similarities*

In the K-Nearest Neighbors regression, although the datasets do not have the same size, the chosen number of neighbors ended with the same number which was 11. This could be explained as both graphs have their peaks when the k is small. As the step was 10 in each loop, this means that number has a tolerance. In another words, the k could be chosen between 6 and 16 in both datasets.

Regarding the random forest, although both datasets have a steady state in performance after 5 decision trees, this does not guarantee that they reached to the most efficient number of decision trees. In fact, the accuracy of random forest regression in the first dataset exceeded the 99% in all runs, while the accuracy in the second dataset didn’t exceed 90% in any run, and this accuracy may need to be improved by consuming more time. In addition, resulted faster outputs when choosing MSE error evaluation criteria in the decision trees’ splitting. This could be explained as the random forest regressor has an efficient algorithm to measure the mean square error faster than the mean absolute error.

### *4.3.2 Differences*

One of the noticeable differences in the results is the performance. Although the second dataset’s size is larger than the first dataset size -which gives any regression model the chance to be more accurate-, the accuracy of the first dataset from all regression models were better than the results of the second dataset accuracy. This could be explained due to the dataset nature, as the output of the second dataset correlation with the features is not high enough, which means that there are other factors that affect the output rather the given features.

In the K-Nearest Neighbors regression, the first dataset resulted better output when the weight calculation criterion was “uniform”, on the other hand, the second dataset resulted better output when the weight calculation criterion was “distance”. As the difference between performance in both criteria in each dataset was not very high, this result could be explained as it does not really matter to choose either uniform or distance criterion in the given datasets.

Regarding the fitting time parameter, the K-Nearest Neighbors recorded as the best model in the first dataset in all runs, while the linear regression model recorded as the best model in the second dataset in most of the runs. As the number of the records in the second dataset is larger than the first dataset, the K-Nearest neighbors regressor has to calculate the Euclidean distance for larger number of points, therefore, the fitting time of the K-Nearest Neighbors regression in the second dataset will be larger than the fitting time in the first dataset. On the other side, the linear regression model was not affected with the difference in the records number. So now it could be clear that if the number of records is large as what happened in the second dataset, the K-Nearest Neighbors regression’s fitting time may exceed the linear regression’ fitting time.

# 5. Conclusions

In conclusion, there are many regression models that could be used to predict a response variable that depend on some features. Each model has its own advantages and disadvantages. By addressing two datasets which hold different information, it could be clear that the results would have some similarities as the same algorithms were applied, and some differences due the dataset nature. Different size and correlation could affect the prediction accuracy regardless regression method. Regarding the evaluation of the models, there are many parameters to be considered such as the implementation time, the fitting time, and the performance. Some models are the best in performance while others are the best in time. Therefore, choosing the best model depends on putting all parameter into consideration to choose the suitable speed and performance which could satisfy the given problem.

**6. References**

1. Leite, R. Introduction to Cross-Validation: K-Fold, Analytics Vidyha (2020).

<https://medium.com/analytics-vidhya/introduction-to-cross-validation-k-fold-7ed9cbd0ed7b>

1. Dutta, A. Random Forest Regression in Python, GeeksforGeeks (2020).

<https://www.geeksforgeeks.org/random-forest-regression-in-python/>

1. Stojiljkovic, M. Linear Regression in Python (2020). <https://realpython.com/linear-regression-in-python/#:~:text=The%20package%20scikit%2Dlearn%20is,learn%20is%20also%20open%20source>.
2. Xifara, A. Energy efficiency Data Set. Machine Learning Repository (2012)

<http://archive.ics.uci.edu/ml/datasets/Energy+efficiency>

1. Cheng Yeh, I. Concrete Compressive Strength Data Set. Machine Learning Repository (2007)

<http://archive.ics.uci.edu/ml/datasets/Concrete+Compressive+Strength>

# 7. Directly used references

1. Dutta, A. Random Forest Regression in Python, GeeksforGeeks (2020).

<https://www.geeksforgeeks.org/random-forest-regression-in-python/>

1. Leite, R. Introduction to Cross-Validation: K-Fold, Analytics Vidyha (2020).

<https://medium.com/analytics-vidhya/introduction-to-cross-validation-k-fold-7ed9cbd0ed7b>