GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

LOCATION ESTIMATION ON MOBILE NETWORKS

Ahmed Hakan KILIÇ

Supervisor
Assist. Prof. Dr. Ali BOYACI

M.Sc. THESIS
COMPUTER ENGINEERING DEPARTMENT
ISTANBUL - 2021

ACCEPTANCE AND APPROVAL PAGE

On 31/08/2021, **Ahmed Hakan KILIÇ** successfully defended the thesis entitled "**Location Estimation on Mobile Networks**" which he prepared after fulfilling the requirements specified in the associated legislations, before the jury members whose **signatures are listed below. This thesis is accepted as a MASTER'S THESIS by Istanbul Commerce University,** Graduate School of Natural and Applied Sciences, **Computer Engineering Department**.

Supervisor	**Assist. Prof. Dr. Ali BOYACI** Istanbul Commerce University	.....
Jury Member	**Assist. Prof. Dr. Alper ÖZPINAR** Istanbul Commerce University	.....
Jury Member	**Assist. Prof. Dr. Zeynep GÜRKAŞ AYDIN** Istanbul University - Cerrahpaşa	.....

Approved Date: 27/09/2021

Istanbul Commerce University, Graduate School of Natural and Applied Sciences, accordance with the 1st article of the Board of Directors Decision dated 27.09.2021 and numbered 2021/322, "Ahmed Hakan KILIÇ" who has determined to fulfill the course load and thesis obligation was unanimously decided to graduate.

Prof. Dr. Necip ŞİMŞEK
Head of Graduate School of Natural and Applied Science

ACADEMIC AND ETHICAL RULES DECLARATION OF CONFORMITY

In this project I prepared in accordance with the rules of thesis writing, Istanbul Commerce University, Institute of Science,

- I obtained all the information and documents in the project within the framework of academic rules.
- I present all visual, audio, and written information and results in accordance with scientific moral rules.
- I refer to the related works in accordance with scientific norms in case of using others' works.
- I cited all the works I cited as a source.
- I did not make any distortions in the data used.
- and that I do not present any part of this project as another thesis study at this university or another university.

I declare.

27/09/2021

Ahmed Hakan KILIÇ

CONTENTS

Page

ABSTRACT

M.Sc. Thesis

LOCATION ESTIMATION ON MOBILE NETWORKS

Ahmed Hakan KILIÇ

Istanbul Commerce University
Graduate School of Applied and Natural Sciences
Department of Computer Engineering

Supervisor: Assist. Prof. Dr. Ali BOYACI
2021, 47 pages

This thesis reports the location estimation on Mobile networks using Base Station (BTS) data. Processed data have been collected from the field as TA (Timing Advance), RSRP (Reference Signal Received Power), and RSRQ (Reference Signal Received Quality) measurements. We also gathered the corresponding Global Positioning System (GPS) to the measurements. Location estimation results compared to the actual location.

Keywords: Base station, GPS, mobile networks, machine learning, RSRP, RSRQ, timing advance (TA).

ÖZET

Yüksek Lisans Tezi

MOBİL AĞLARDA LOKASYON TAHMİNİ

Ahmed Hakan KILIÇ

İstanbul Ticaret Üniversitesi
Fen Bilimleri Enstitüsü
Bilgisayar Mühendisliği Anabilim Dalı

Danışman: Dr. Öğr. Üyesi Ali BOYACI
2021, 47 sayfa

Bu tez, Baz İstasyonu (BTS) verilerini kullanarak Mobil ağlarda konum tahminini raporlamaktadır. İşlenen veriler sahadan TA (Timing Advance), RSRP (Reference Signal Received Power) ve RSRQ (Reference Signal Received Quality) ölçümleri olarak toplanmıştır. Ölçümlere karşılık gelen Global Konumlandırma Sistemi (GPS) verileri de toplanmıştır. Lokasyon tahminleri gerçek lokasyonla (GPS) karşılaştırılmıştır.

Anahtar Kelimeler: Baz istasyonu, GPS, mobil ağlar, makine öğrenmesi, RSRP, RSRQ, timing advance (TA).

ACKNOWLEDGEMENTS

Firstly, I would like to thank my supervisor Assist. Prof. Dr. Ali BOYACI for his help with his knowledge during the development of the research.

I would also like to thank my family and friends for their never-ending support, prayers, and patience.

Ahmed Hakan KILIÇ
ISTANBUL, 2021

FIGURES

Page

TABLES

Page

SYMBOLS AND ABBREVIATIONS

AOA	Angle Of Arrival
ARD	Automatic Relevance Determination Regression
BTS	Base Station
dB	Decibel
dbm	Decibel Milliwatts
GPS	Global Positioning System
Lars	Least-angle regression
Lasso	Least Absolute Shrinkage And Selection Operator
MAE	Mean Absolute Error
MSE	Mean Squared Error
MSLE	Mean Squared Logarithmic Error
RMSE	Root Mean Squared Error
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
TA	Timing Advance
TDOA	Time Difference of Arrival

1. INTRODUCTION

Location estimation on Mobile networks is becoming an essential topic in recent years. Whether it is an emergency, improving service quality, or providing an alternative way of location estimation. BTS data can be used to locate the users connected to it. BTS does not feature a location-providing service, but we can manipulate and fine-tune the data they provide to estimate the user's location. There are several ways to estimate the location of the user using BTS data.

Whether using Microcell Zone concept Samarah (2016), using TDOA and AOA measurements with Nelder-Mead algorithm Lee, et al. (2019), applying Kalman Filtering or triangulation method which data from three BTS is used to triangulate the coordinates of the user relative to the BTS Anisetti, et al. (2011). Although the triangulation method seems the easiest way of locating user, it has some hard aches. BTS are not located in a similar geographic location, and they should be configured in the environment they placed to cover all the areas to improve the service quality. The data gathered from the rural areas may be different from a crowded city center.

In our approach, we used actual data gathered from the field. There is no need for special hardware or updates on the BTS. BTS are not dedicated to providing location service, and we just used the data already available on the BTS. Our methodology is based on the geographical position of three BTS and triangulation. In addition to triangulation, we also used the GPS coordinates corresponding to the measurement data we have. We had an opportunity to compare our result with the actual location.

We located the users connected to the BTS, and by doing this we may be able to increase the service quality. Configuration of the BTS for the environment can further improve estimating the location of the user. BTS can improve the service quality by locating the user's location.

The remainder of this paper is organized as follows. Literature review, data and problem, solution and methodology and conclusion sections.

2. LITERATURE REVIEW

The document in Samarah (2016), presented a location estimation of a user in a Microcell Zone Concept (MZC) Mobile Station by retrieving TA from the BTS. In MZC, two or more zone sites are connected to the same BTS. This gives an advantage of the user being served with the strongest signal within the zone. When the user moves from one zone to a different zone Mobile Station Controller (MSC) changes the channel to the zone.

The user will remain in the same frequency, so there will be no need for a handoff procedure. The study conducted using MATLAB simulation. But in the real environment, results may be adversely affected due to geographic location or BTS requiring more sophisticated changes to change the channel from one zone.

Lee, et al. (2019), also performed simulations for using TDOA and AOA measurements using Nelder-Mead (NM algorithm). They proposed a method that is a combination of TDOA and AOA to improve the accuracy of location estimation. They applied NM algorithm to reduce the environmental error to enhance the accuracy of location estimation. They confirmed effectiveness with simulation results, but in real environments such as crowded locations, the results may be affected.

Anisetti et al. (2011), proposed to identify possible locations from signals using the Database Correlation Method (DCM). They then described a technique to deal with signal fluctuations to select paths with greater accuracy. Kalman filtering is also applied to gather information about roads and build a path to improve mobility approximation. They also used actual data collected from the field.

The paper in Martinez Hernández et al. (2019), presented a system to locate user's location in an emergency. They used an android application to extract data. Their method uses known positions of BTS. But the application developed may not be compatible with all Android devices.

Martinez Hernández et al, (2019), investigated the usage of supervised machine learning algorithms to outdoor user localization. Their method is based on collecting uplink transmission on the network side at Remote Radio Head (RRH). They used sector-based processing for better Angle of Arrival (AOA) information. Results were evaluated by using Linear Regression, Weighted K-nearest neighbors and Multi-layer perception.

In our approach, we used actual data gathered from the field. We also collected the Global Positioning System (GPS) coordinates corresponding to these measurements. Then we compared the algorithm results with the corresponding GPS coordinates to the measurements.

3. DATA AND PROBLEM

Data we gathered from BTS are TA (Timing Advance), RSRP (Reference Signal Received Power), and RSRQ (Reference Signal Received Quality). We also collected the GPS coordinates corresponding to these measurements. By just using these data, the average distance to the BTS cannot be found. These measurements are correlated with the distance; we did fine-tune and used these measurements in the distance calculation.

TA (Timing Advance) is the time that it takes to a signal to reach from the user's device to the BTS.

BTS calculates the delay of the data, and if it sees a delay, it increases the TA value by 1. The maximum TA value is 63. This TA value can tell the BTS how far the user is from it. Each level of TA is 550 meters.

RSRP (Reference Signal Received Power) is the measurement of the power of the signal. RSRP value is measured on dbm type.

RSRQ (Reference Signal Received Quality) is signals quality, which tells us how signal compared to noise. We may have enough power (RSRP), but the quality will be lower if we also have noise in the system. RSRQ value is measured in dB.

We also applied the formula 3.1. for calculating the TA. We calculated the First TA and Last TA by using formula in 3.1., x being First TA or Last TA. Taking advantage of these, we trained several machine learning algorithms.

$$\left(\frac{X}{16} \times 78\right) + 8 \tag{3.1}$$

When the calculation is made according to the TA formula, the user is within the radius. We tried to find the location by reducing the radius of this circle. The tolerance can be seen in the yellow-colored area on the fig. 1. The yellow-colored area is the margin; we want it to be more precise.

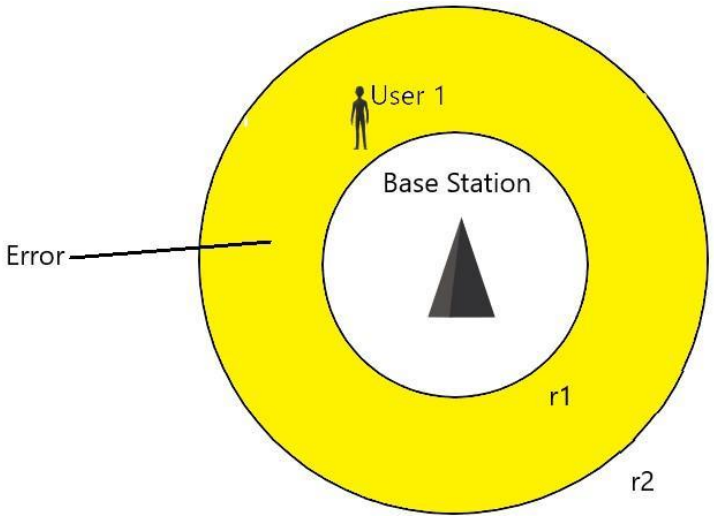

Figure 3.1 Error and margin presentation

R1 is the radius of the BTS, and the distance between R1 and R2 is the error margin. Users can be anywhere in the yellow-colored area. It comes at regular intervals; we can find a sharper location using these intervals to make it more precise than a circle.

User connection transfer from one BTS to the next BTS is called handover. By doing triangulation using BTS data, we can estimate the location of the user.

The problem here is every BTS has different physical conditions, different geographical structures, hills, pits, weather conditions, demographic density, and roads affect the signal. One universal model will not be sufficient to solve this problem because the situation in each BTS is different. In our approach, we used different models for each BTS.

4. METHODOLOGY

4.1 Methodology

We used different models for each BTS. While creating these models, we used eight machine algorithms. We used more than one machine learning algorithm because every BTS has different conditions, as mentioned Data and Problem section. These Machine algorithms are Gaussian, KNeighbor, Linear Regression, ARD, Lars, Lasso, Ridge and Bayesian.

Data harvested from the downtown part of a big city in Turkey. It is a crowded city environment. Before training these machine learning algorithms, we performed data cleaning by removing 31 BTS out of 397. The removed BTS data contained some outlier values and was affecting the result of the machine learning algorithms. The machine learning algorithms are run for all 366 distinct BTS.

We evaluated the results of these machine learning algorithms using Means Absolute Error, Mean Squared Error, Median Absolute Error, and R^2.

The details of the machine learning algorithms, regression metrics and histograms figures are in the following section.

4.2 Machine Learning Algorithms

Machine learning algorithms is a computer science where it uses data and various algorithms to learn and improve its accuracy. We briefly explained the machine learning algorithms used on this thesis.

4.2.1 Gaussian process regression

Gaussian probability distribution functions are the summary of the distribution of random variables, Gaussian processes summarize the properties of the functions. Gaussian processes can be used as a machine learning algorithm for classification predictive modeling. (Ruan, et al., 2017) The formula of Gaussian Process Regression is presented on 4.1.

$$p(F_N|X_N)\frac{\exp(-\frac{1}{2}F_N^T K_N^{-1}F_N)}{\sqrt{(2\pi)^N \det(K_N)}} \quad (4.1)$$

4.2.2 KNeighbor regression (KNN)

KNN regression is a method which approximates the relationship between independent variables and the continuous outcome by averaging the observations in the same neighborhood.

In KNN classification output is the membership of the neighborhood in regression it's the value of the object. is the property value of the object. This value is the average of the values of its nearest neighbors. (Hirose, et al., 2021)

Formula of KNeighbor Euclidean presented on 4.2. it is the square root of the sum of differences between x and y. KNeighbor Manhattan formula on presented on 4.3.it is the absolute values of difference between x and y. KNeighbor Minkowski formula presented on 4.4. where parameter q is either 1 or 2. If q is equal to 1 it is Manhattan distance, if 2 it is Euclidean.

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \quad (4.2)$$

$$\sum_{i=1}^{k}|x_i - y_i| \quad (4.3)$$

$$\left(\sum_{i=1}^{k}|x_i - y_i|\right)^{1/q} \quad (4.4)$$

4.2.3 Linear regression

Regression determines the relationship between one dependent variable and several other independent variables. Linear regression is used to find the best fit straight line or hyperplane for a set of points. In other words, linear regression establishes a relationship between the dependent variable and one or more independent variables using the best fit straight line. (Hirose, et al., 2021) Linear Regression formula is written in the form of Y = a+bX where X is independent, and Y is the dependent variable.

$$A = \frac{[(\sum y)(\sum x^2) - (\sum x)(\sum xy)]}{[n(\sum x^2) - (\sum x)^2]} \tag{4.5}$$

$$B = \frac{[n(\sum xy) - (\sum x)(\sum y)]}{[n(\sum x^2) - (\sum x)^2]} \tag{4.6}$$

4.2.4 Automatic relevance determination regression (ARD)

ARD is very similar to Bayesian Ridge Regression, but it effectively prunes away redundant features. It achieves this by regularizing the solution space with a parameterized data driven distribution. (Van, et al., 2001)

$$p(w|a) = \prod_i N(w_i|0, a_i^{-1}) \tag{4.7}$$

4.2.5 Least angle regression lasso (LARS)

LARS Lasso is implemented using the LARS algorithm where it is used with high dimensional data where it finds the correlated property to the original value. It also averages and finds most efficient direction without overfitting if it finds more than one correlated property to the original value. (Efron, et al., 2004)

4.2.6 Least absolute shrinkage and selection operator (LASSO)

Lasso regression is similar to Ridge regression. Lasso regression plays an important role not only in reducing overlearning, but also in feature selection. It's used for used

for classification and prediction. (Li, et al., 2010) The formula is presented on 4.8. where increase of λ means bias increase, decrease of λ means variance increase.

$$\sum_{i=1}^{n}(y_i - \sum_j x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \quad (4.8)$$

4.2.7 Ridge regression

Ridge Regression is obtained by adding a regularization term to our cost function in linear regression. With this addition, the learning algorithm both learns the data and tries to keep the model weights as small as possible. (Li, et al., 2020) The formula of Ridge Regression is presented on 4.9. where λ is penalty term and left side of the equation is regression calculation, and right side is the sum of the β value squared.

$$\sum_{i=1}^{n}(y_i - \bar{y}_i)^2 + \lambda \sum_{j=1}^{P} \beta_j^2 \quad (4.9)$$

4.2.8 Bayesian ridge regression

Bayesian Ridge Regression estimates a probabilistic model of the regression problem. Its aim is not to find best value of the model parameters. It's to determine the posterior distribution for the model parameters. Not only is the response generated from a probability distribution, but the model parameters are assumed to come from a distribution as well. (Pereira, et al., 2020) The formula is presented on 4.10. where w and λ are estimated jointly during the fit of the model.

$$p(w|\lambda) = N(w|0, \lambda^{-1}I_p) \quad (4.10)$$

4.3 Regression metrics

In regression you cannot use classification accuracy to evaluate results, to error metrics must be used to evaluating the results. (Divyanshu, 2019) There are several metrics used for regression. We briefly explained them in the following sections.

4.3.1 Explained variance

Explained variance is used to measure the inconsistency between a model and actual data. It answers the question “how well does this model work?” by giving us a percentage of the explained variance. Higher percentage indicates you make better predictions. (Rosenthal, 2011)

The formula of Explained variance and result of our algorithms can be seen in below histograms. The Gaussian algorithm performed poorly compared to others. The other algorithms had similar performance. Results of our algorithms can be seen in below histograms.

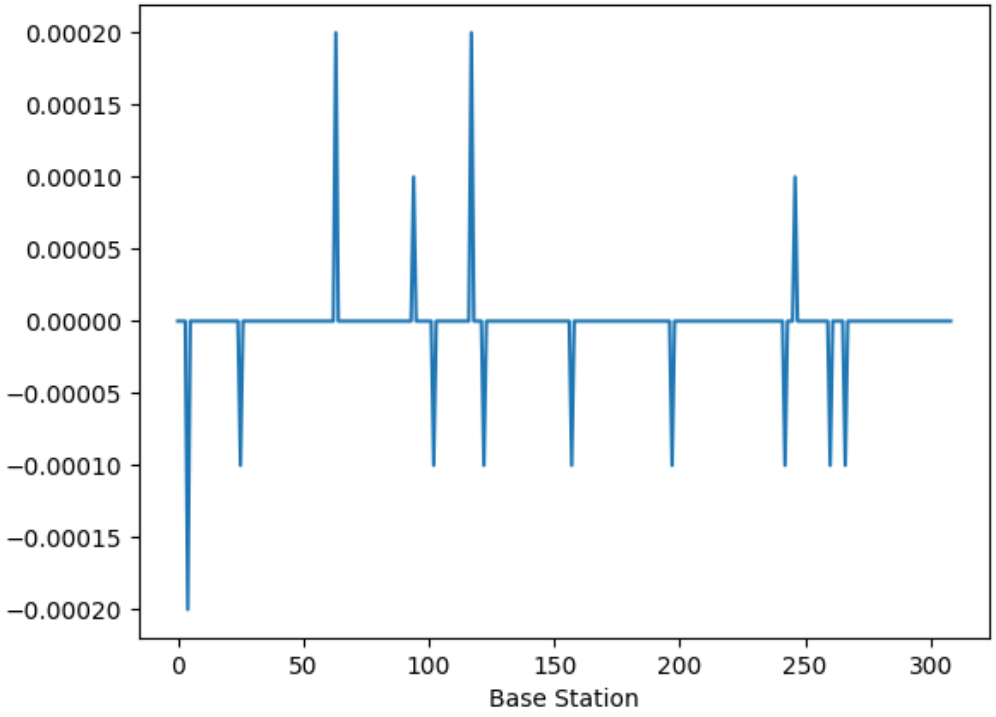

Figure 4.1 Explained variance gaussian processor graph

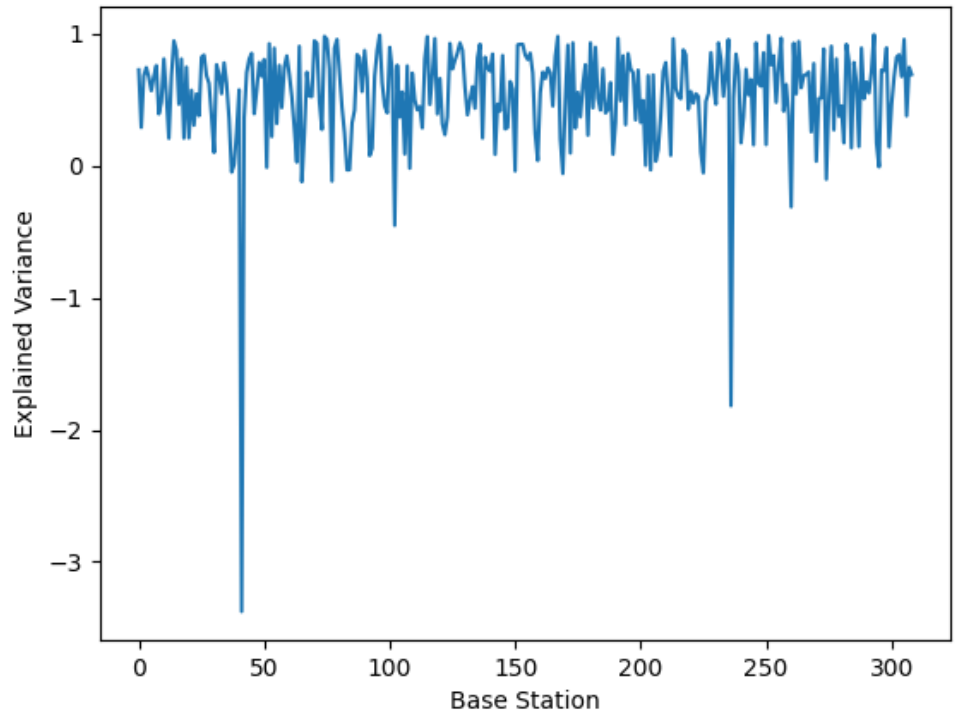

Figure 4.2 Explained variance kneighbor regressor graph

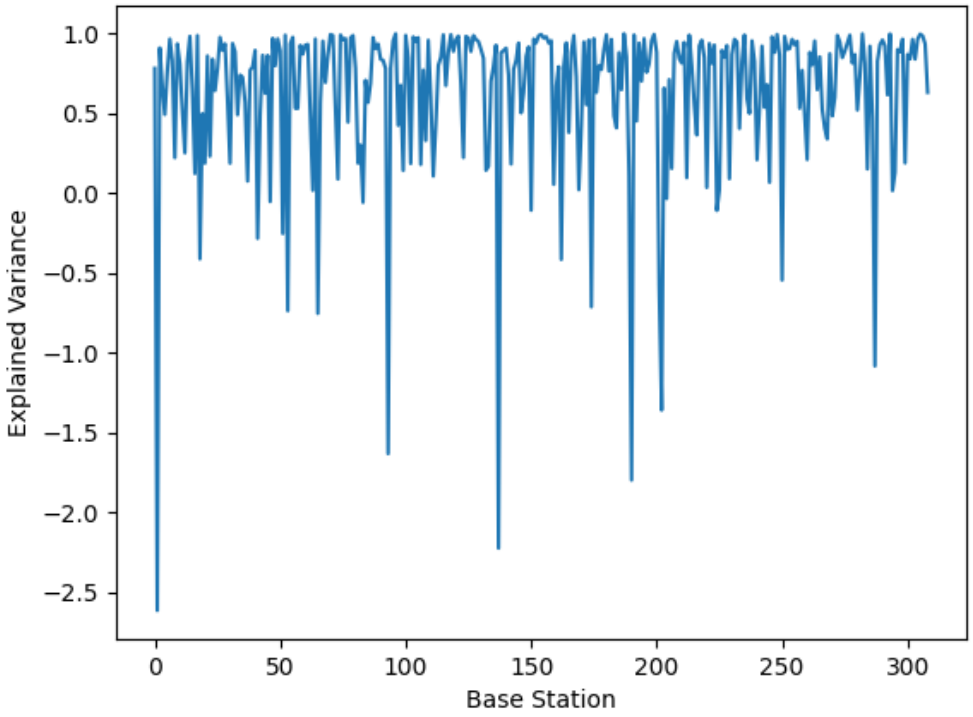

Figure 4.3 Explained variance linear regression graph

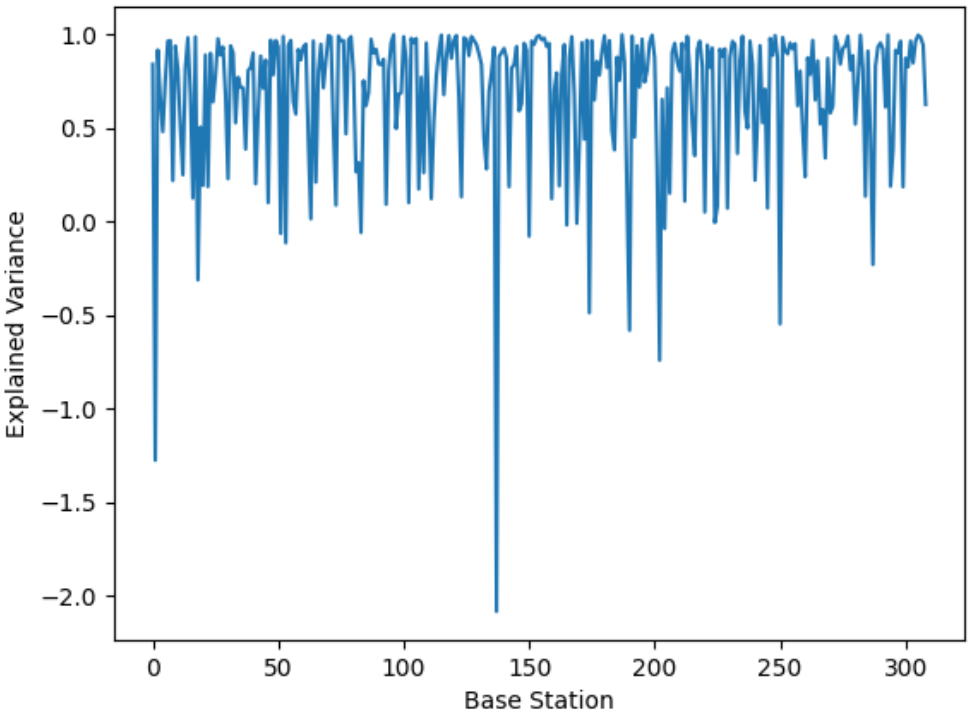

Figure 4.4 Explained variance ard regression graph

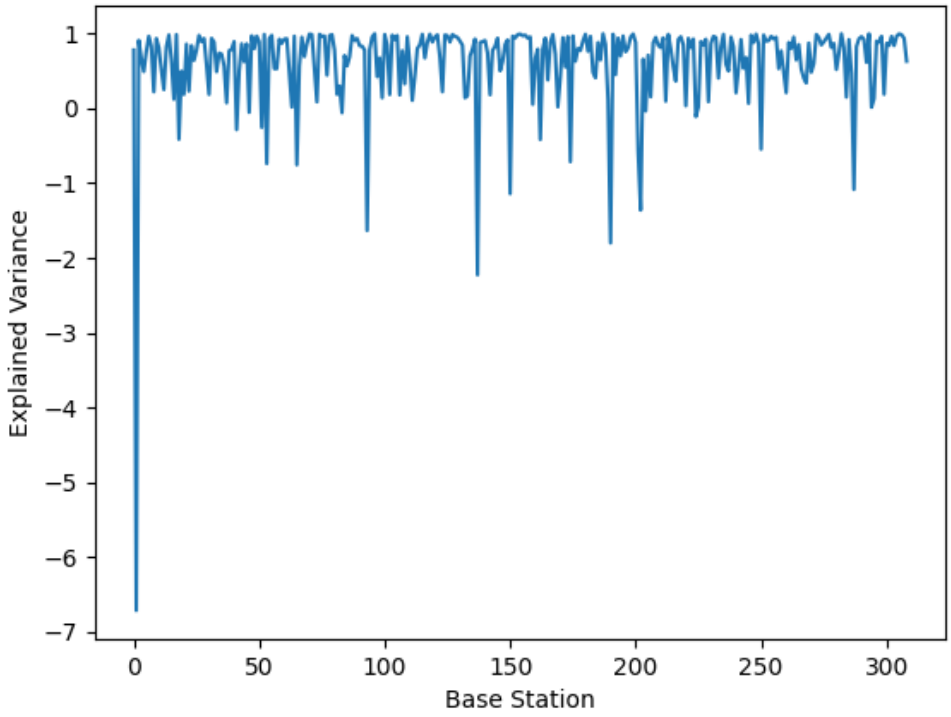

Figure 4.5 Explained variance lars graph

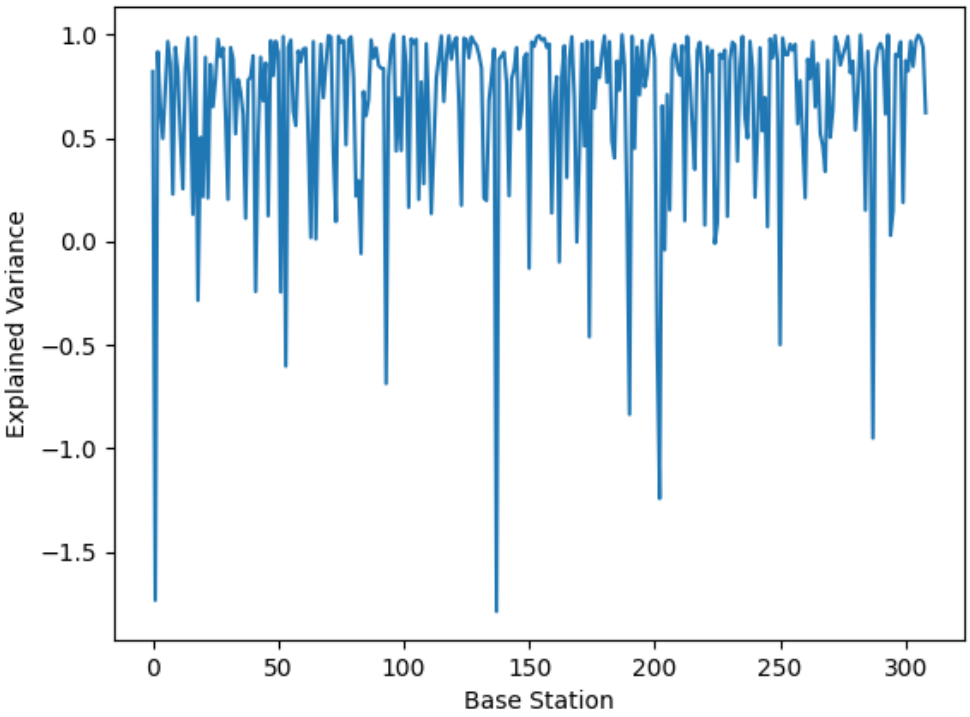

Figure 4.6 Explained variance lasso graph

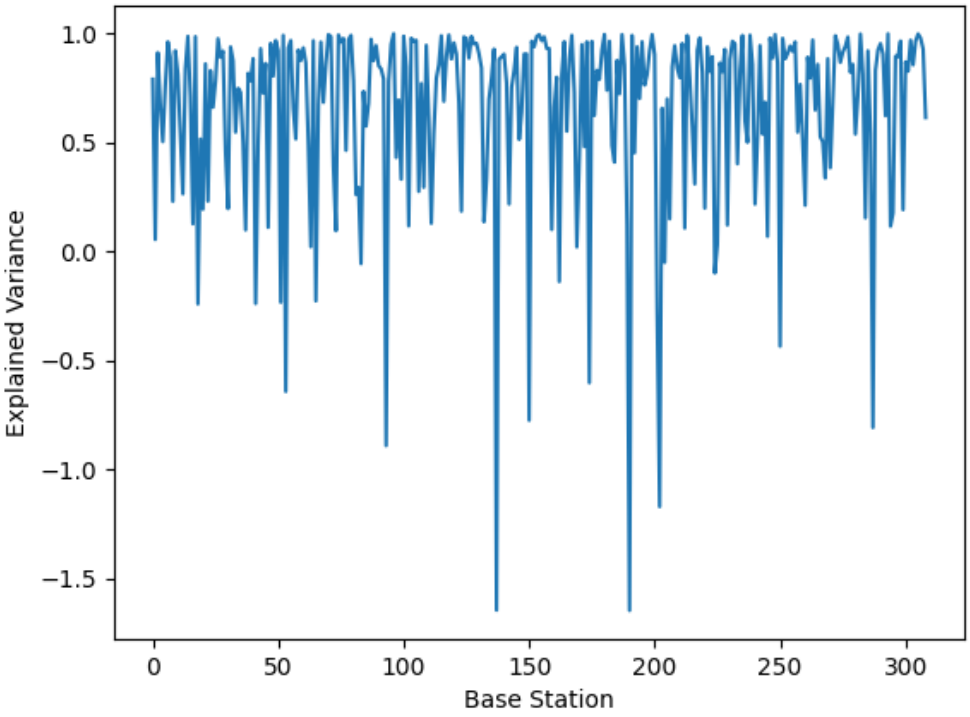

Figure 4.7 Explained variance ridge graph

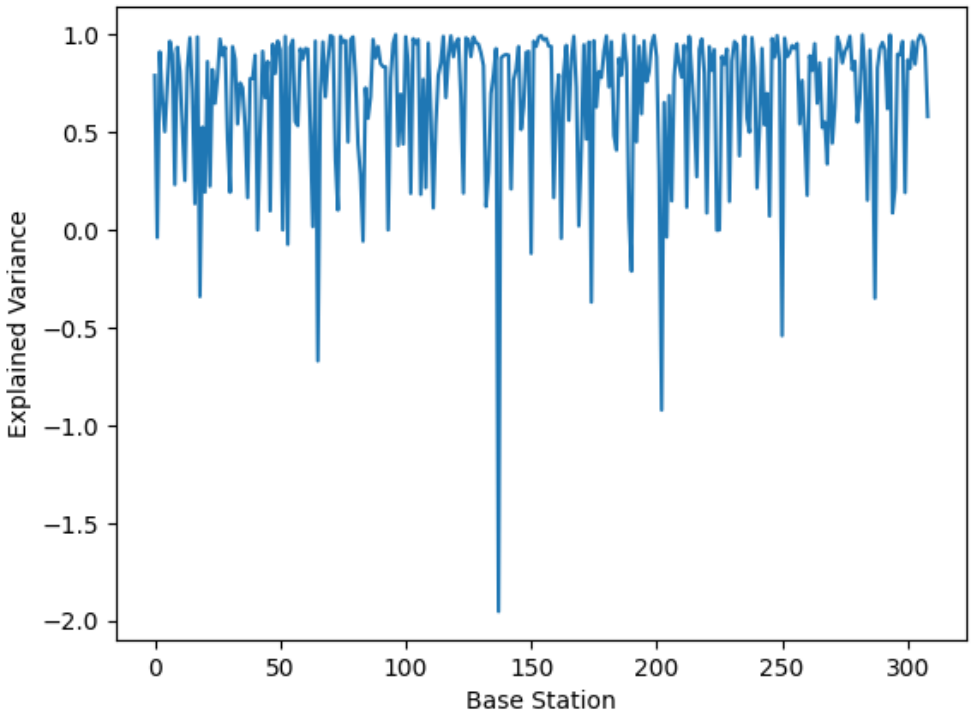

Figure 4.8 Explained variance bayesian ridge regression graph

This graphs for Explained Variance, which shows the fit of the model to the data. It can be seen on the graph that variance is moderately high which means that our data is moderately diverse.

4.3.2 Mean squared logarithmic error (MSLE)

Mean squared logarithmic error can be explained as calculation of ratio between original and predicted values. It takes log of the original and predicted values. It is a variation of the mean squared error.

In MSLE relative difference between original and predicted is important. Using MSLE with regression helps avoiding large errors to be punished compared to small errors.

The formula of MSLE and result of our algorithms can be seen in below histograms. The Gaussian and Bayes algorithm performed poorly compared to others. The other algorithms had similar performance. The formula can be seen on 4.12. where $\hat{y}$ is the predicted value. Results of our algorithms can be seen in below histograms.

$$MSLE = \frac{1}{n}\sum_{i=0}^{n}(\log(y+1) - \log(\hat{y}_i + 1))^2 \quad (4.12)$$

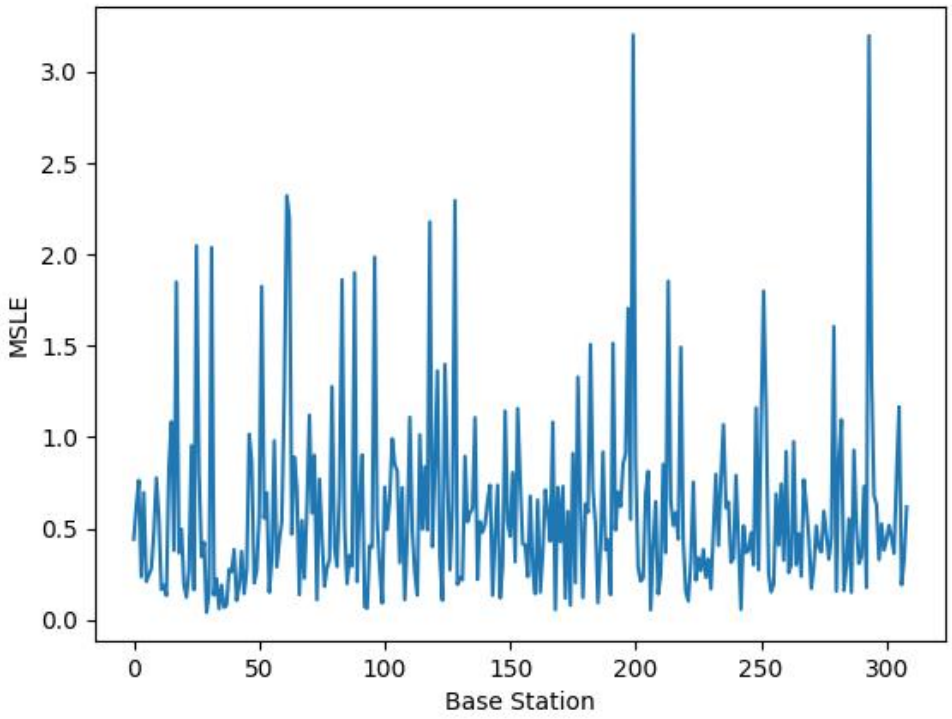

Figure 4.9 Mean squared logarithmic error gaussian processor graph

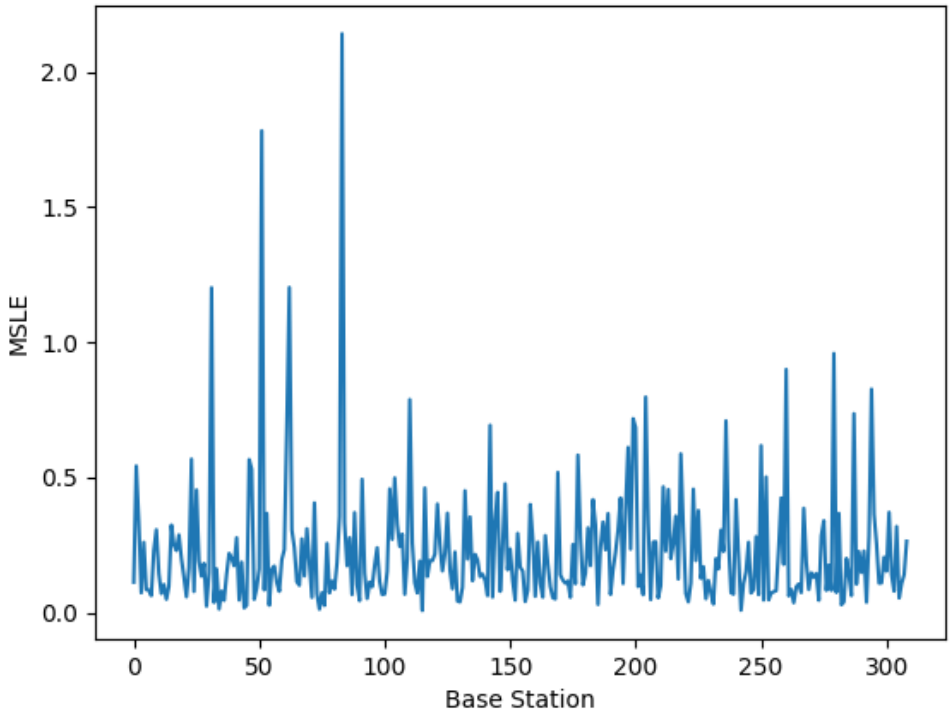

Figure 4.10 Mean squared logarithmic error kneighbor regressor graph

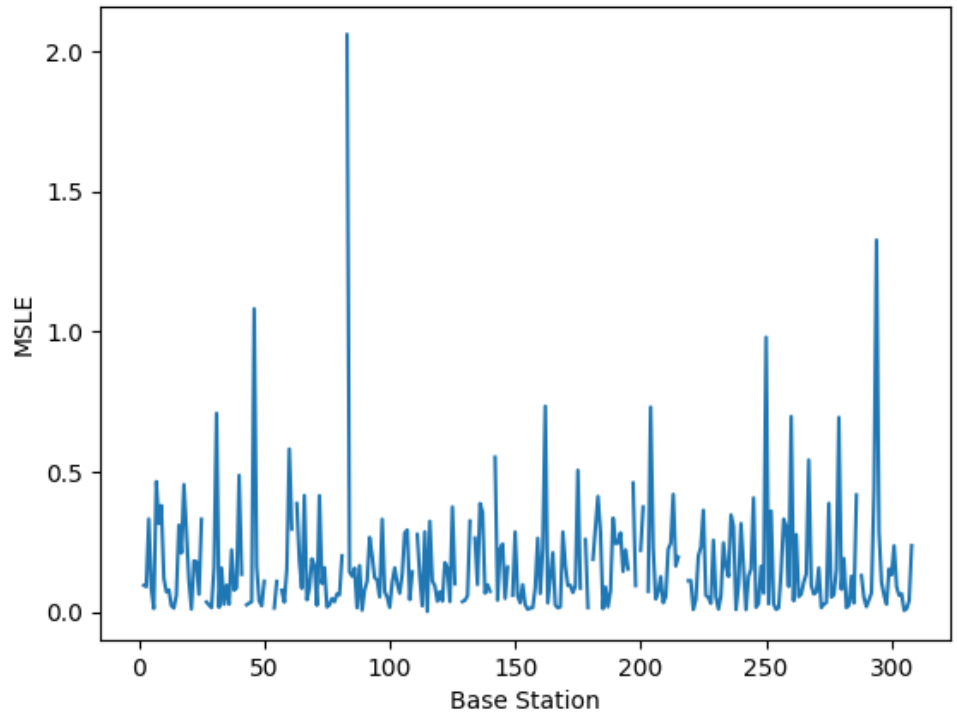

Figure 4.11 Mean squared logarithmic error linear regression graph

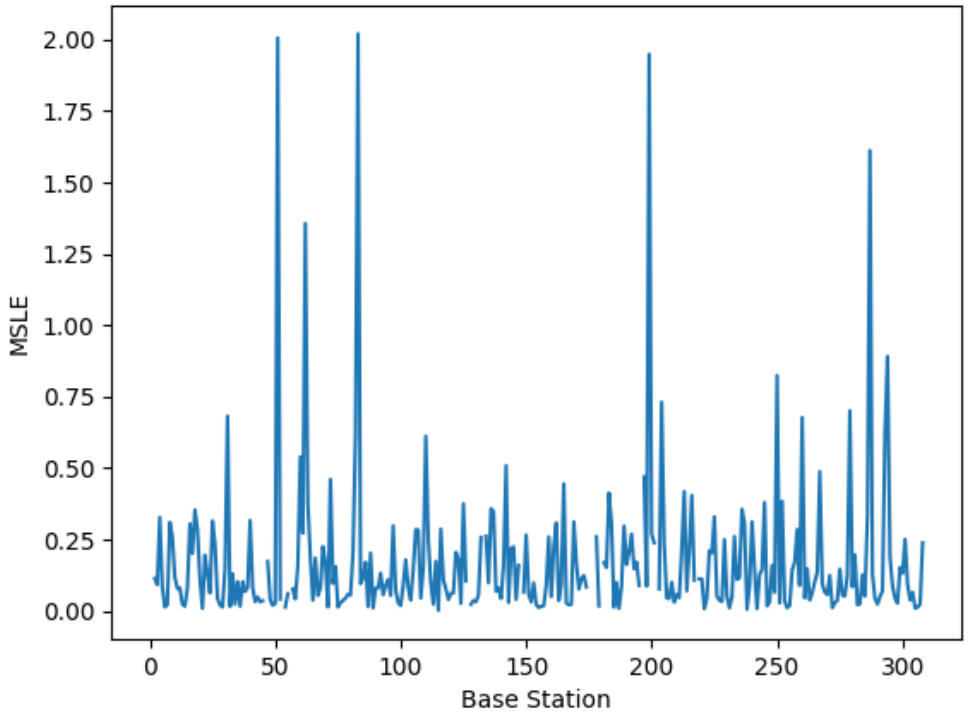

Figure 4.12 Mean squared logarithmic error ard regression graph

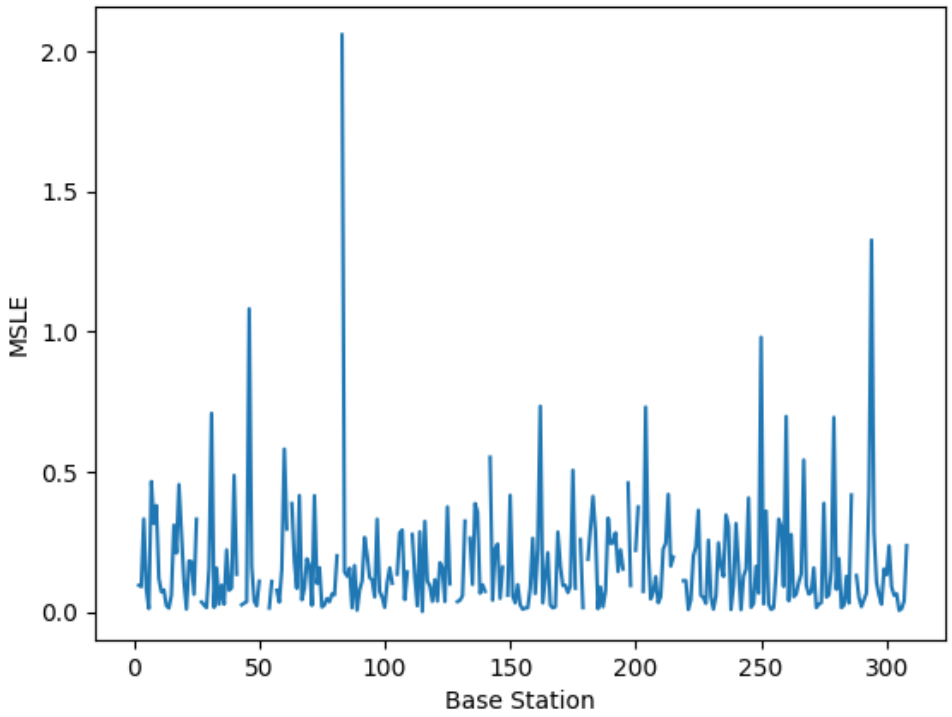

Figure 4.13 Mean squared logarithmic error lars regression graph

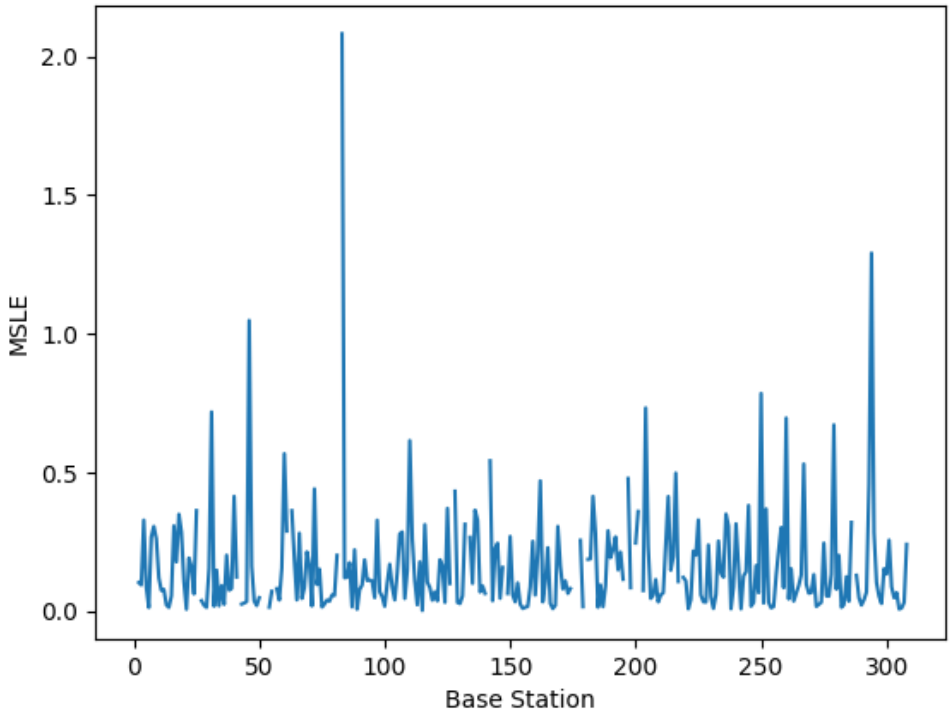

Figure 4.14 Mean squared logarithmic error lasso graph

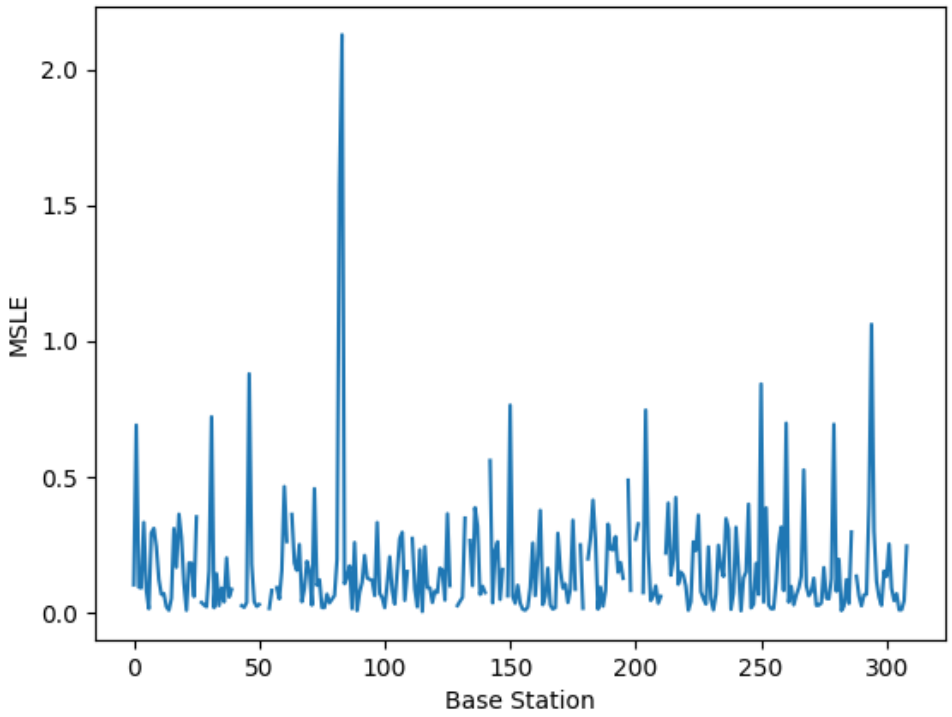

Figure 4.15 Mean squared logarithmic error ridge regression graph

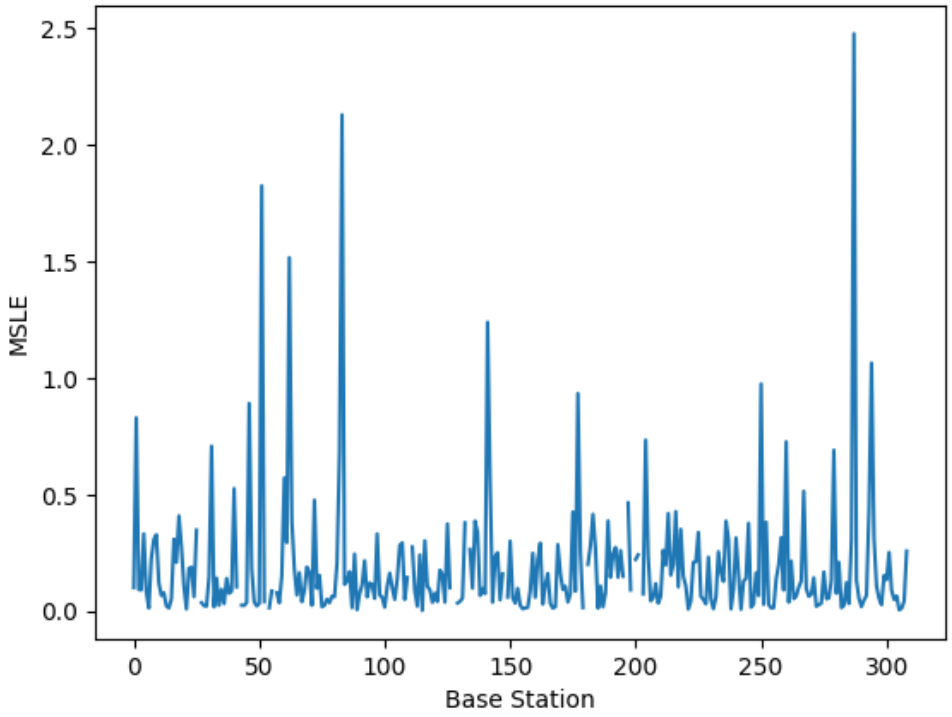

Figure 4.16 Mean squared logarithmic error bayesian ridge regression graph

The graphs for MSLE which shows the ratio between the true and predicted values. It can be seen on the graphs MSLE results are good with minor exceptions on some of the BTS. Overall results are great.

4.3.3 R^2 error

R^2 error is used for evaluating the goodness of fit. Greater the value of R^2 better the regression model fit.

The formula of R2 error can be seen on 4.13. where the upper side of the equation is the unexplained variation, and the below side is total variation. Results of our algorithms can be seen in below histograms.

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \hat{y})^2} \tag{4.13}$$

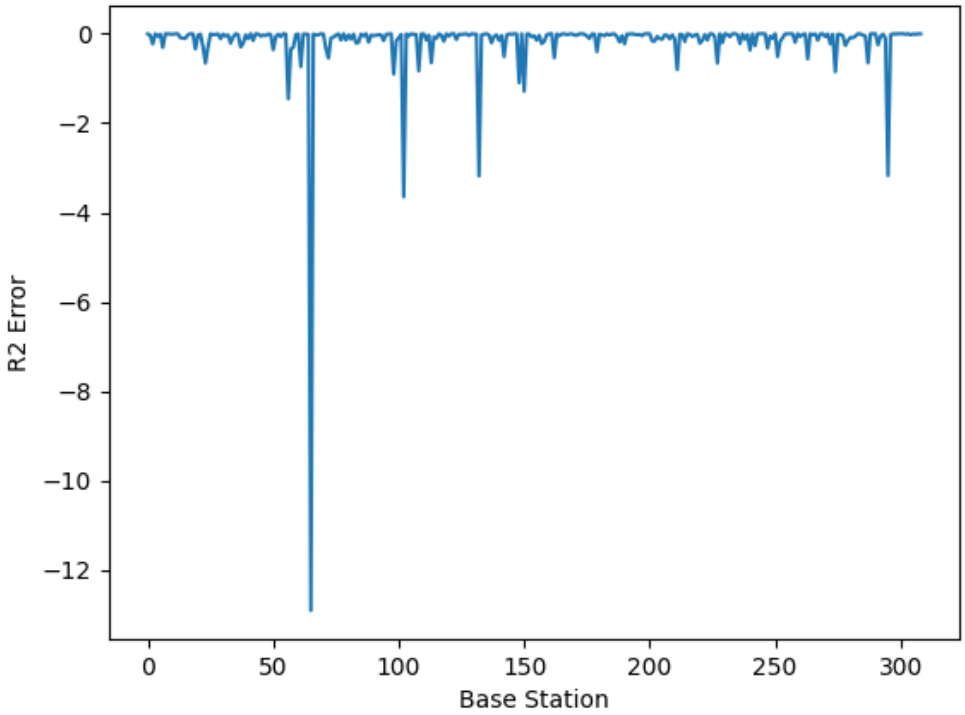

Figure 4.17 R^2 error gaussian processor graph

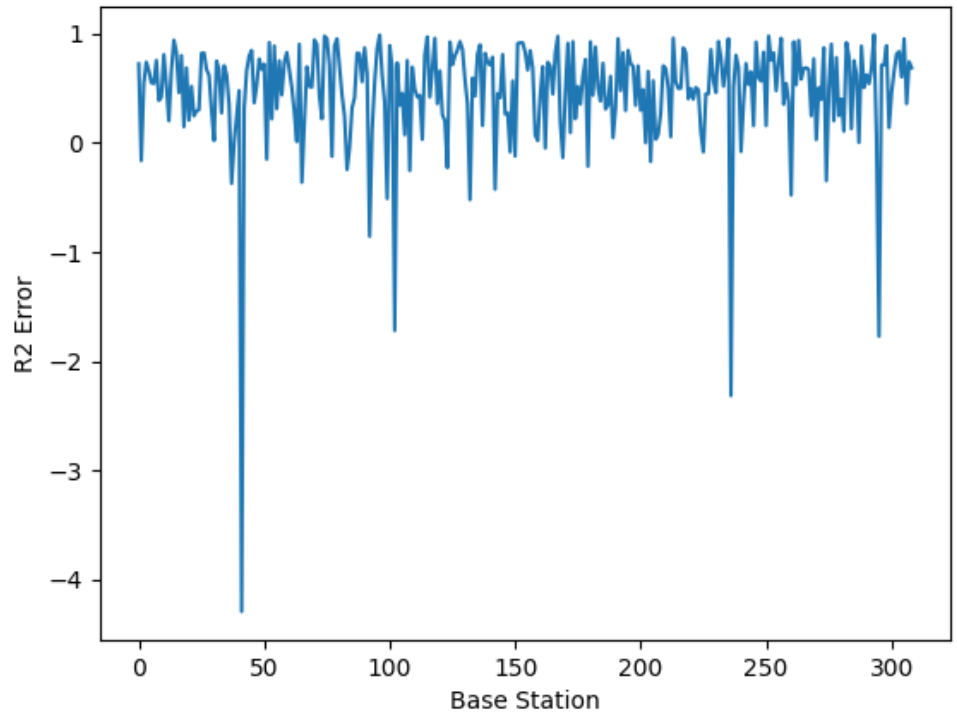

Figure 4.18 R^2 error kneighbor regressor graph

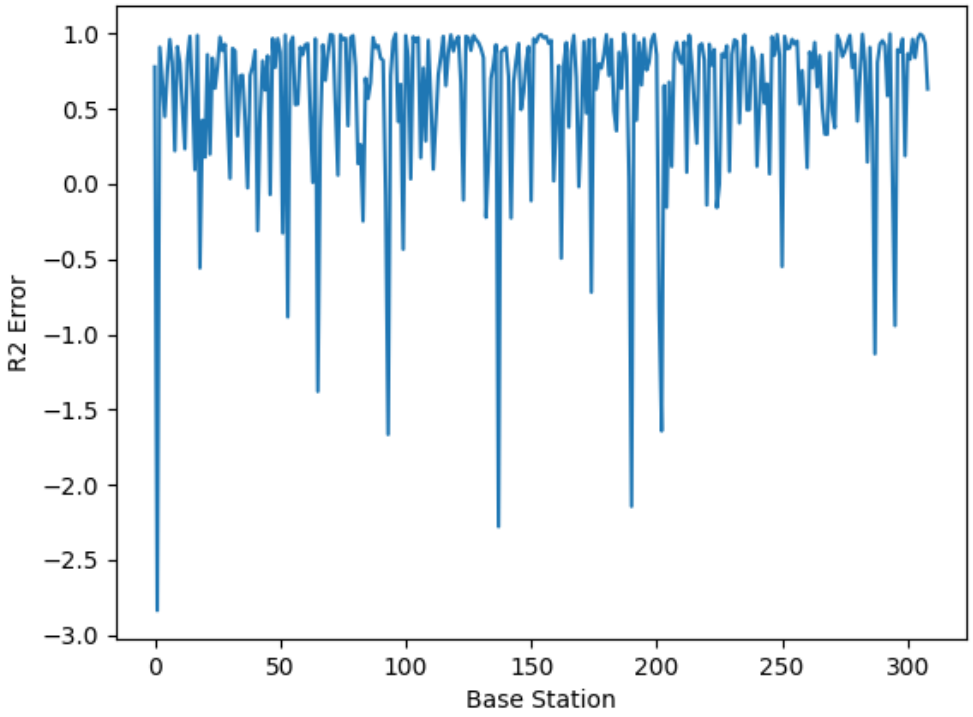

Figure 4.19 R^2 error linear regression graph

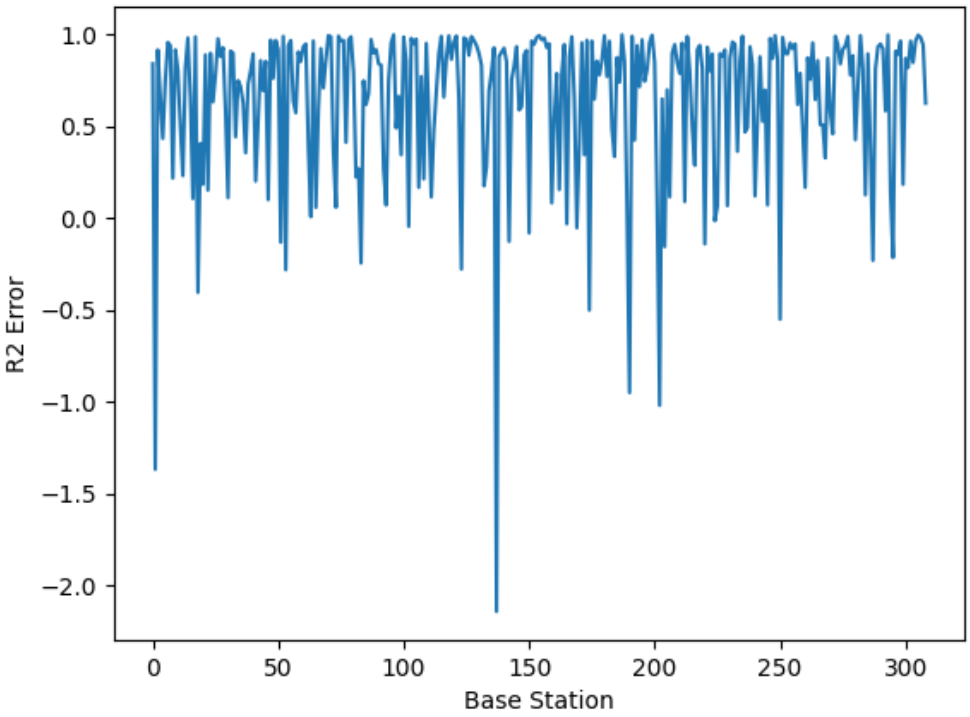

Figure 4.20 R^2 error ard regression graph

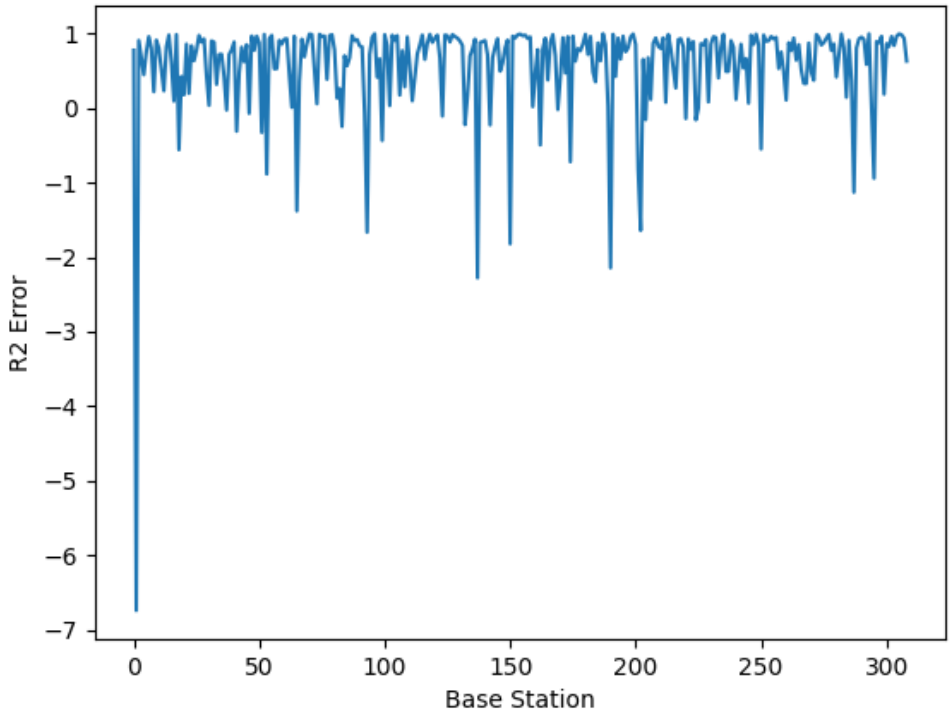

Figure 4.21 R^2 error lars regression graph

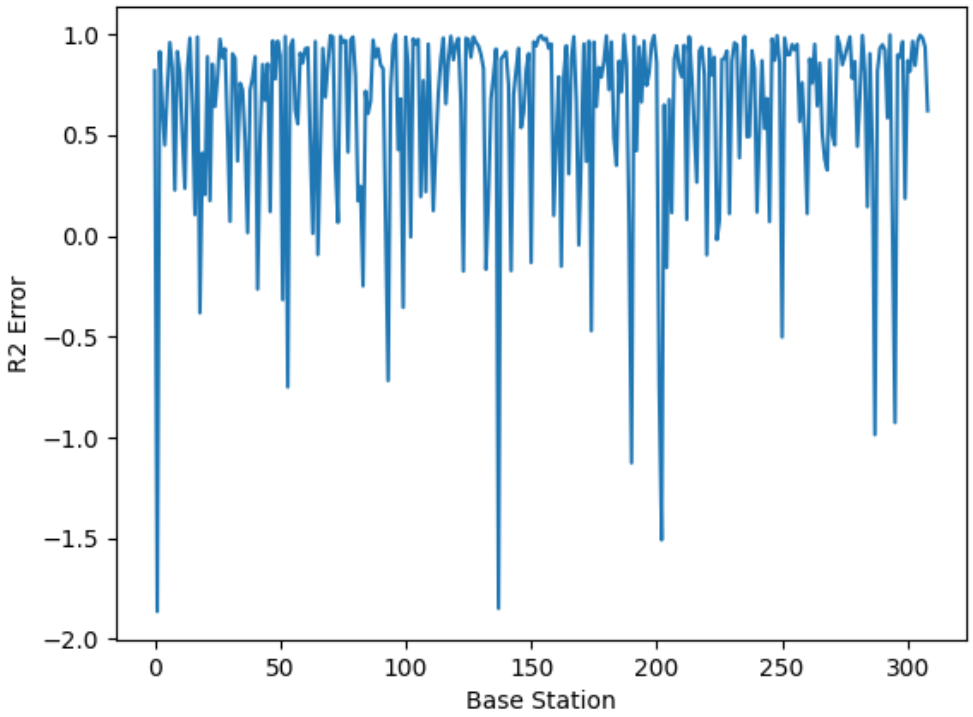

Figure 4.22 R^2 lasso regression graph

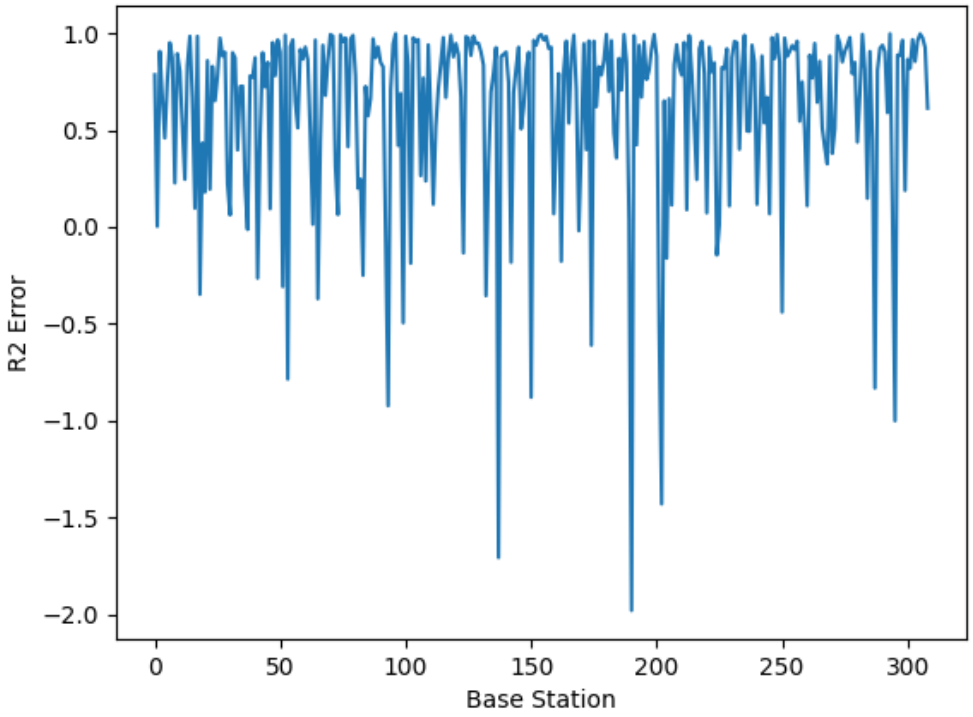

Figure 4.23 R^2 error ridge regression graph

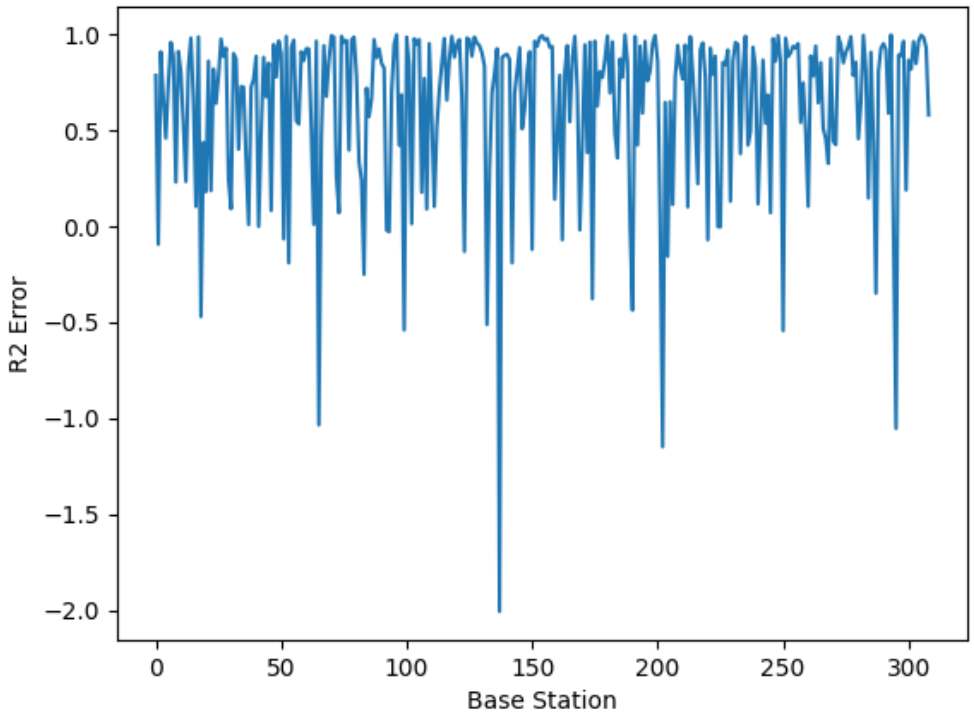

Figure 4.24 R^2 error bayesian ridge regression graph

The R^2 Error graphs are similar to Explained Variance graphs. In graphs the correlation can be seen.

4.3.4 Mean absolute error (MAE)

Mean Absolute Error is the absolute difference between the original value and predicted value. MAE is not as eager to penalize the errors as MSE.

The formula of MAE is on 4.14. where n is the numbers of errors and $|x_i - x|$ is the absolute errors. Results of our algorithms can be seen in below histograms.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |x_i - x| \quad (4.14)$$

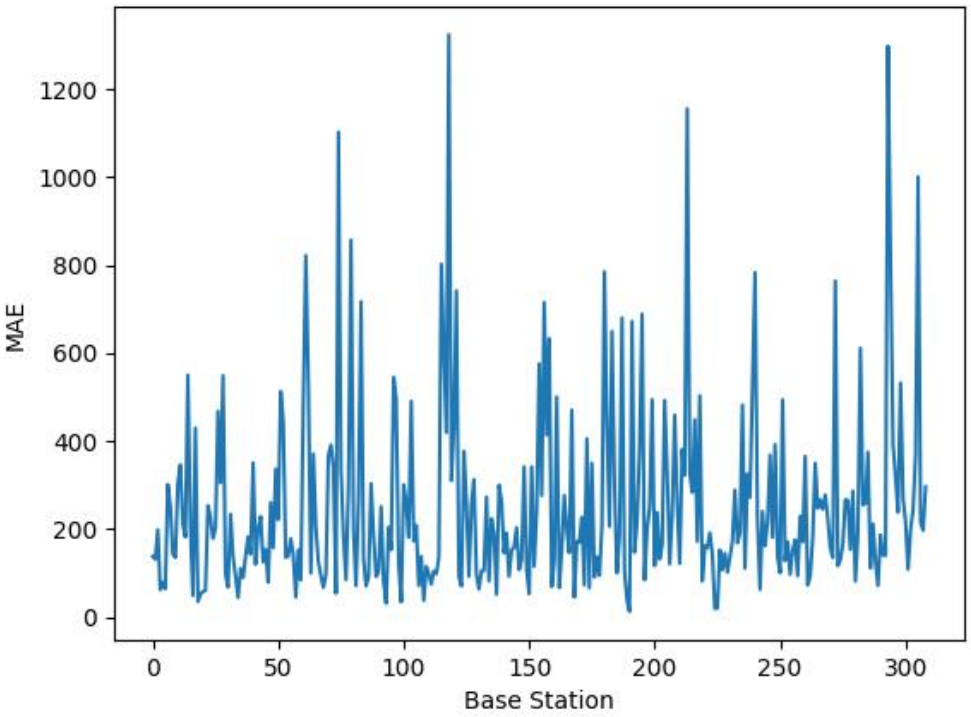

Figure 4.25 Mean absolute error gaussian processor graph

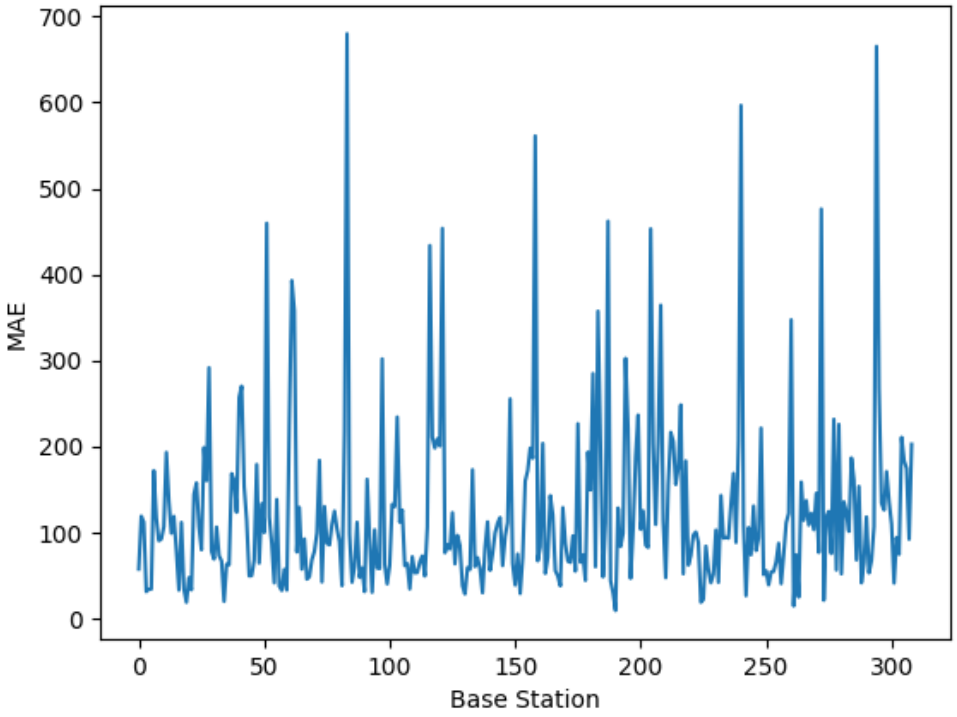

Figure 4.26 Mean absolute error kneighbor regressor graph

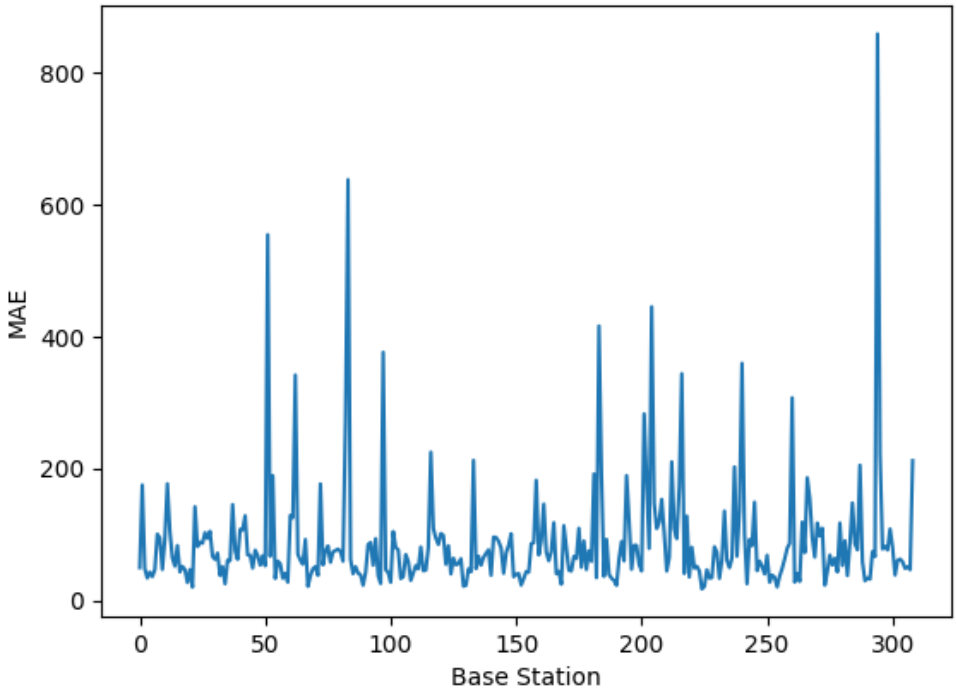

Figure 4.27 Mean absolute error linear regression graph

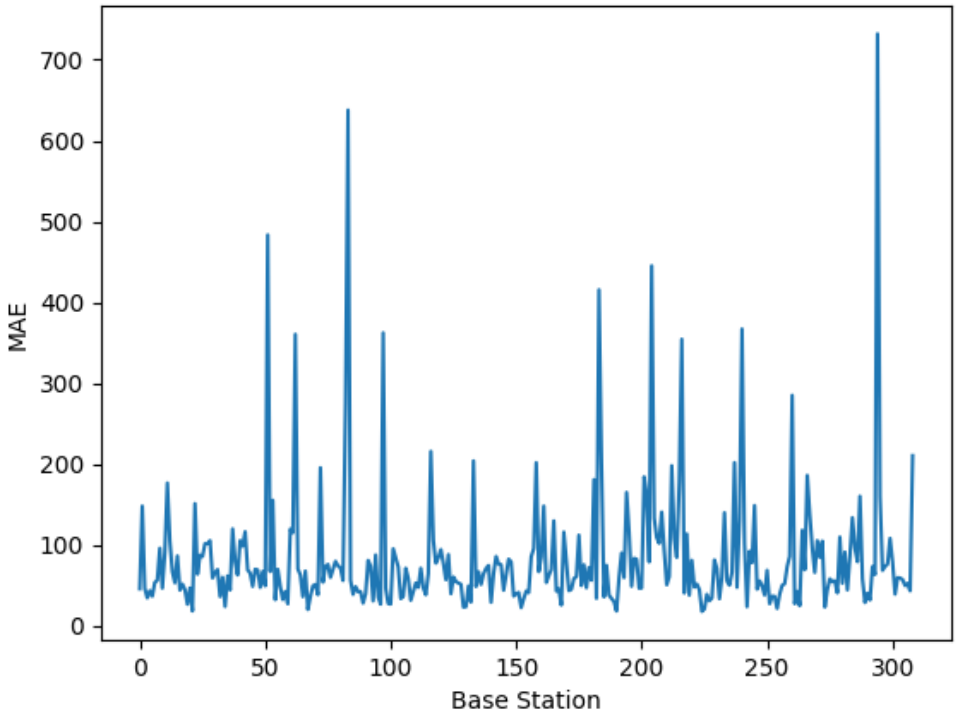

Figure 4.28 Mean absolute error ard regression graph

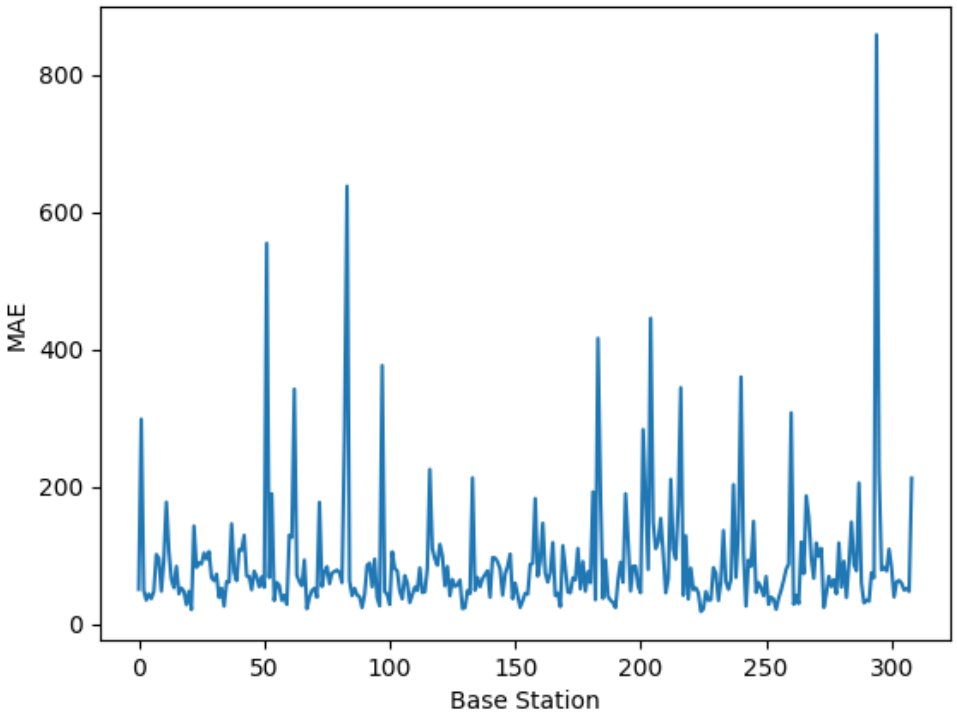

Figure 4.29 Mean absolute error lars regression graph

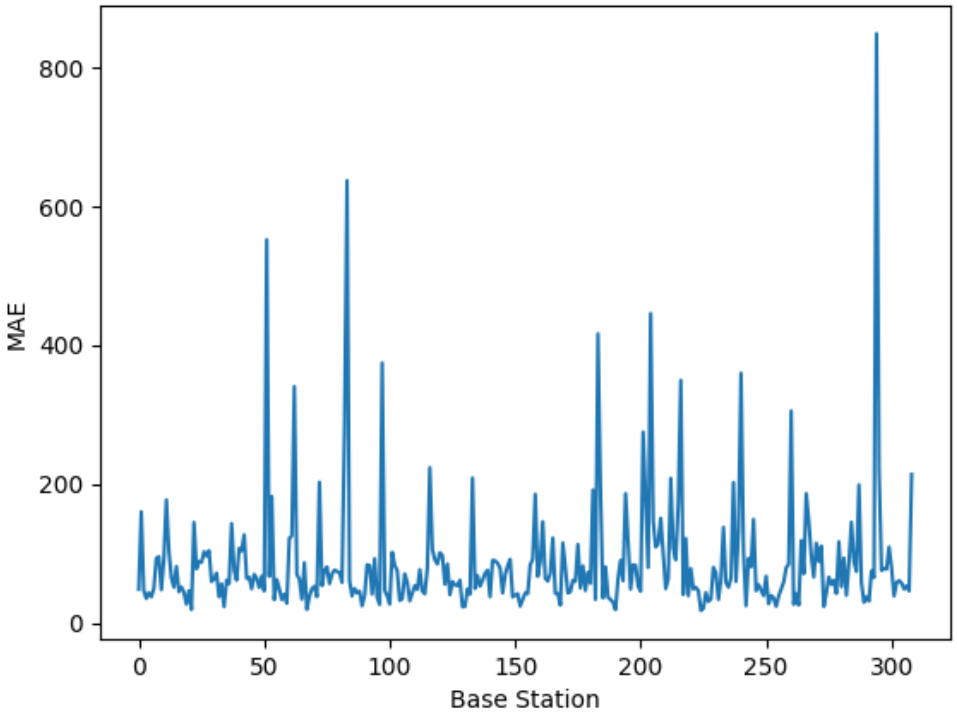

Figure 4.30 Mean absolute error lasso regression graph

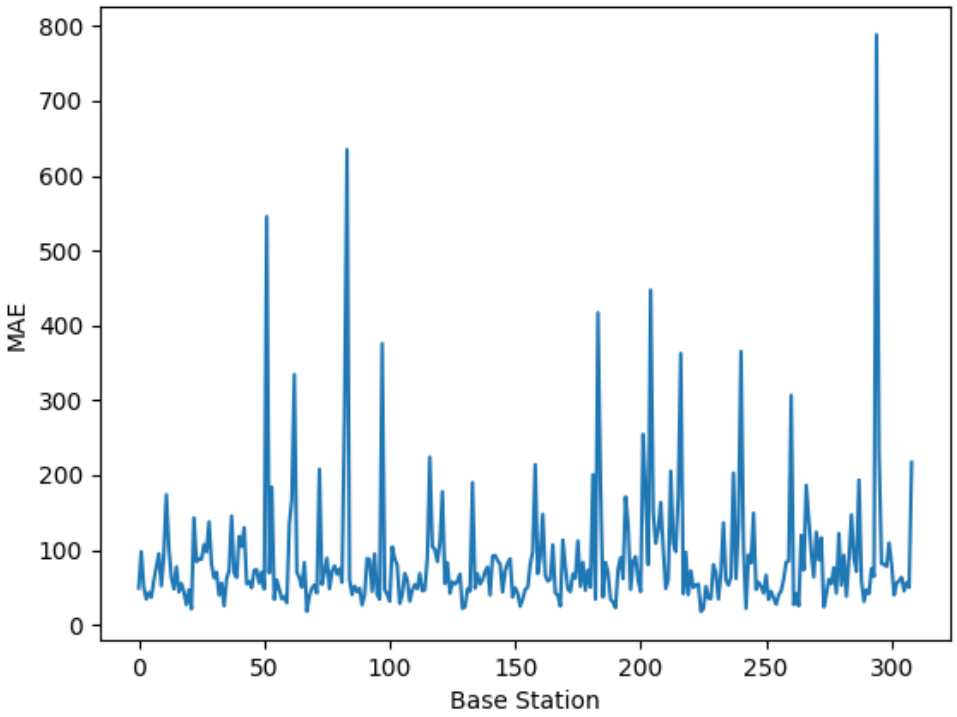

Figure 4.31 Mean absolute error ridge regression graph

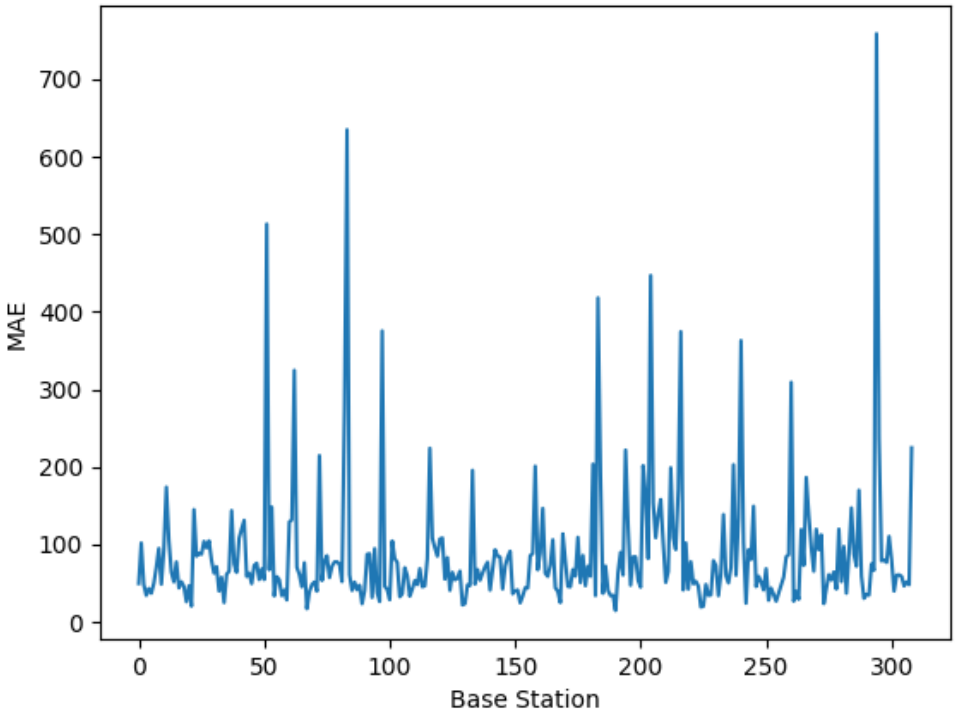

Figure 4.32 Mean absolute error bayesian ridge regression graph

The graph for MAE shows the ratio between the true and predicted values. It can be seen on the graphs MSLE results are good with minor exceptions on some of the BTS.

4.3.5 Mean squared error (MSE)

MSE is average of the squared difference between value aimed value and value produced by the algorithm. It measures the performance of a machine learning model. The result may be negative or positive. If the MSE value is closer to the zero better the performance of the model. The formula of MSE can be seen on 4.15. and $(y - \hat{y})^2$ is the square difference between original and predicted values. Results of our algorithms can be seen in below histograms.

$$MSE = \frac{1}{n}\sum(y - \hat{y})^2 \tag{4.15}$$

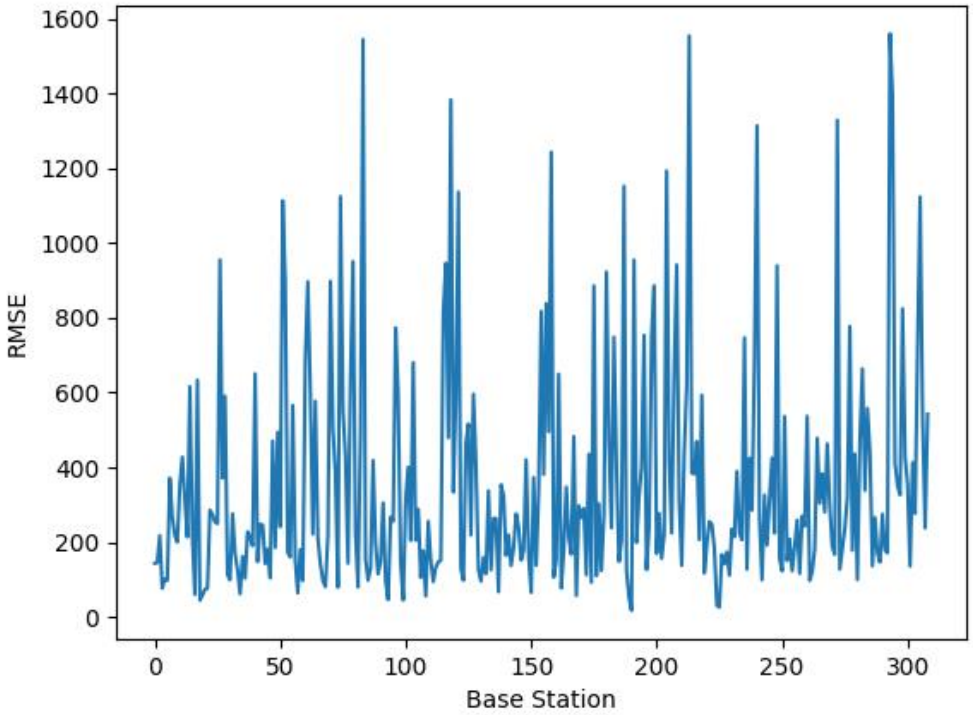

Figure 4.33 Mean squared error gaussian processor graph

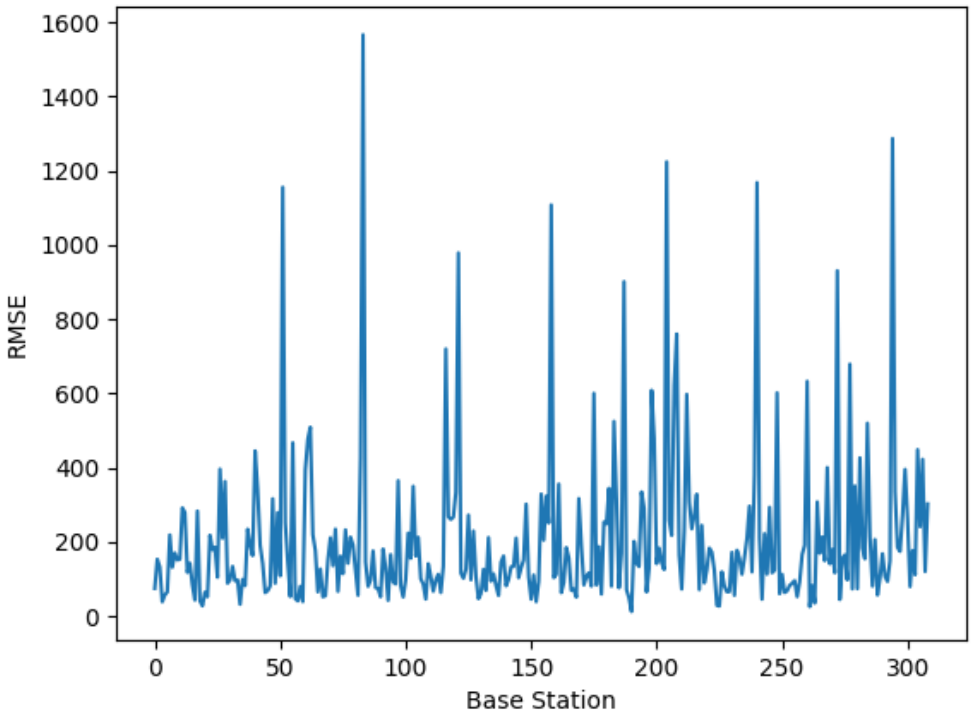

Figure 4.34 Mean squared error kneighbor regressor graph

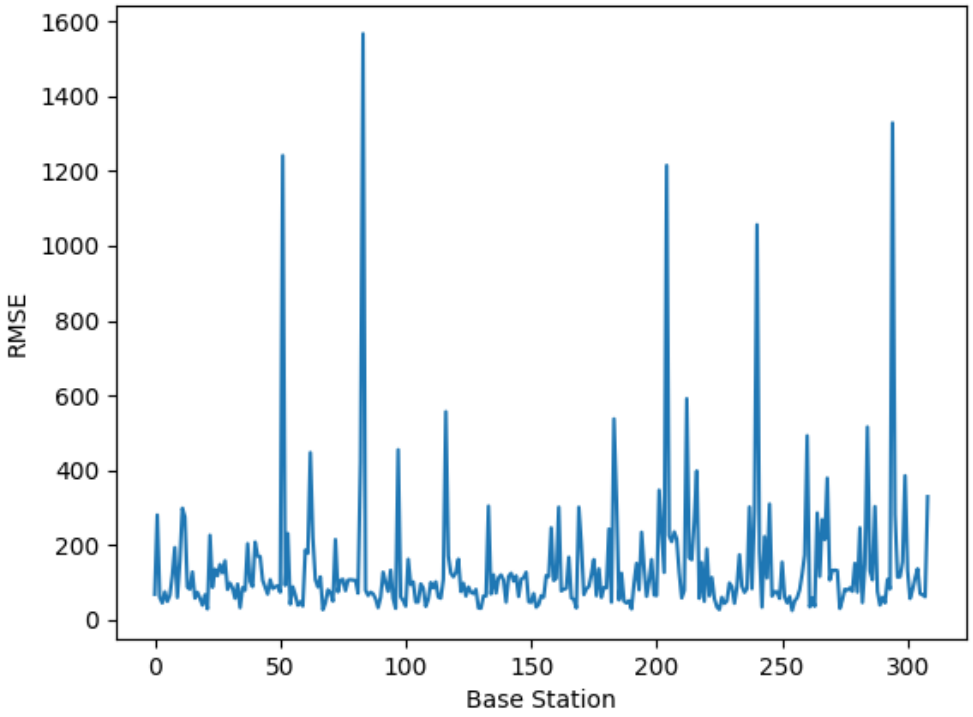

Figure 4.35 Mean square error linear regression graph

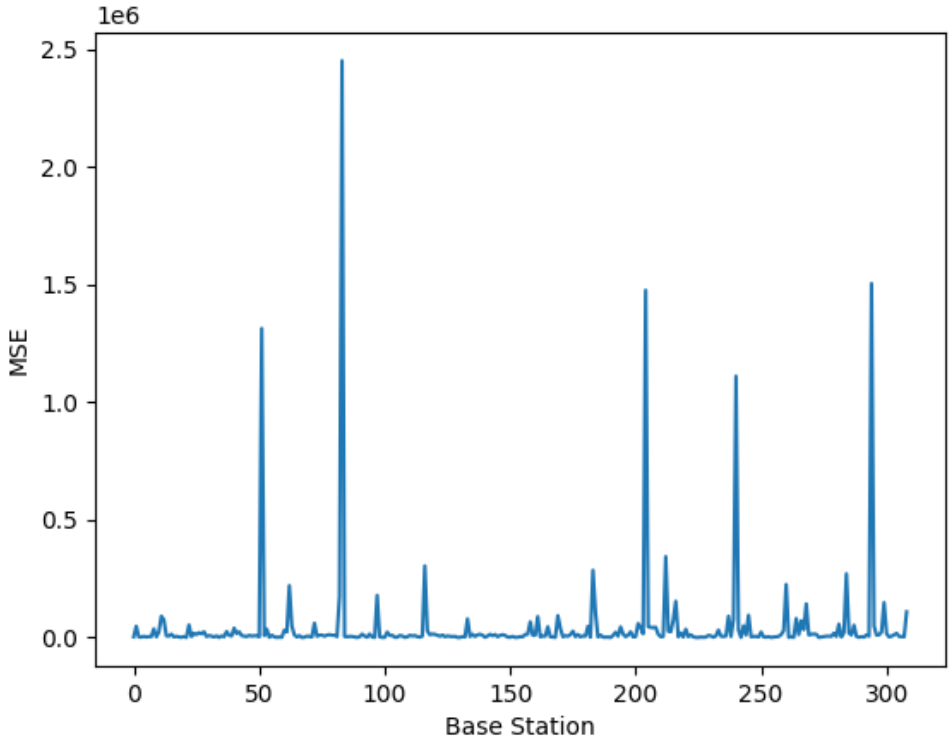

Figure 4.36 Mean squared error ard regression graph

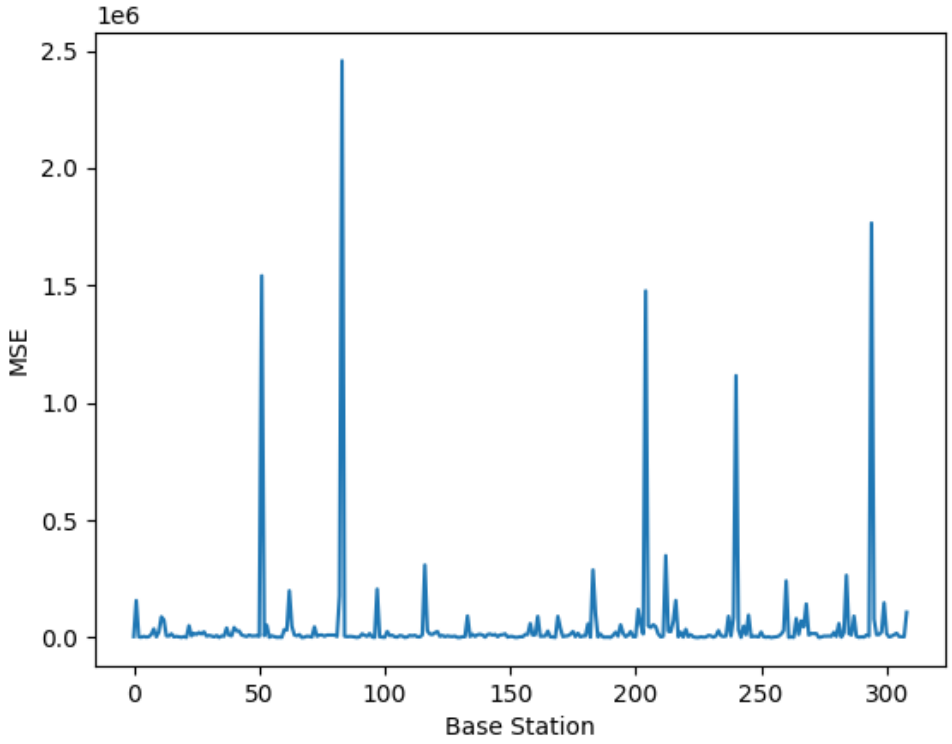

Figure 4.37 Mean squared error lars regression graph

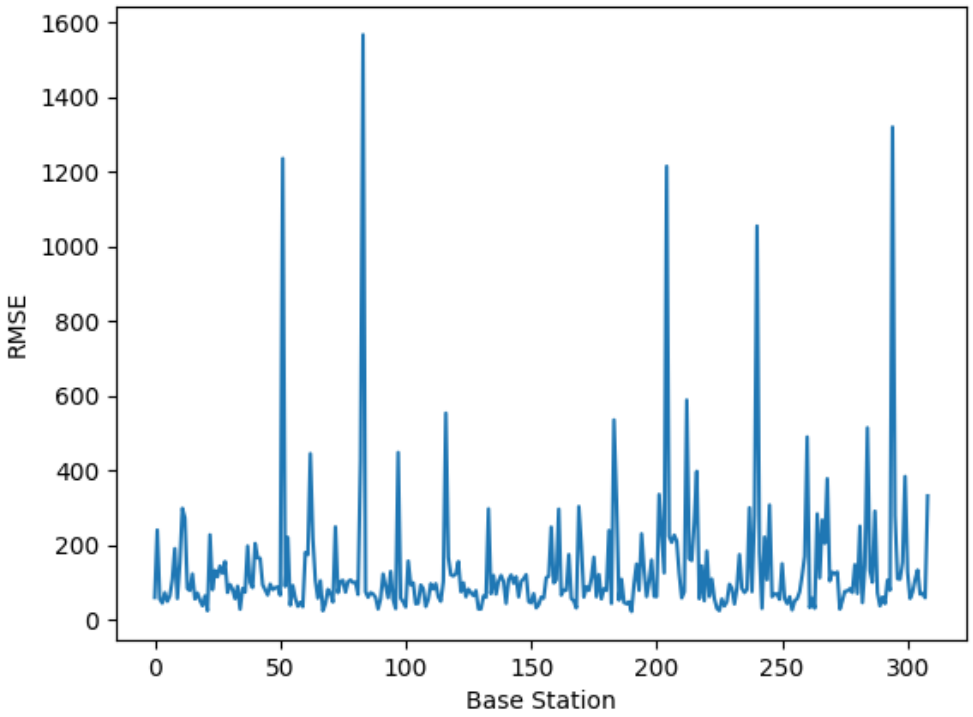

Figure 4.38 Mean squared error lasso regression graph

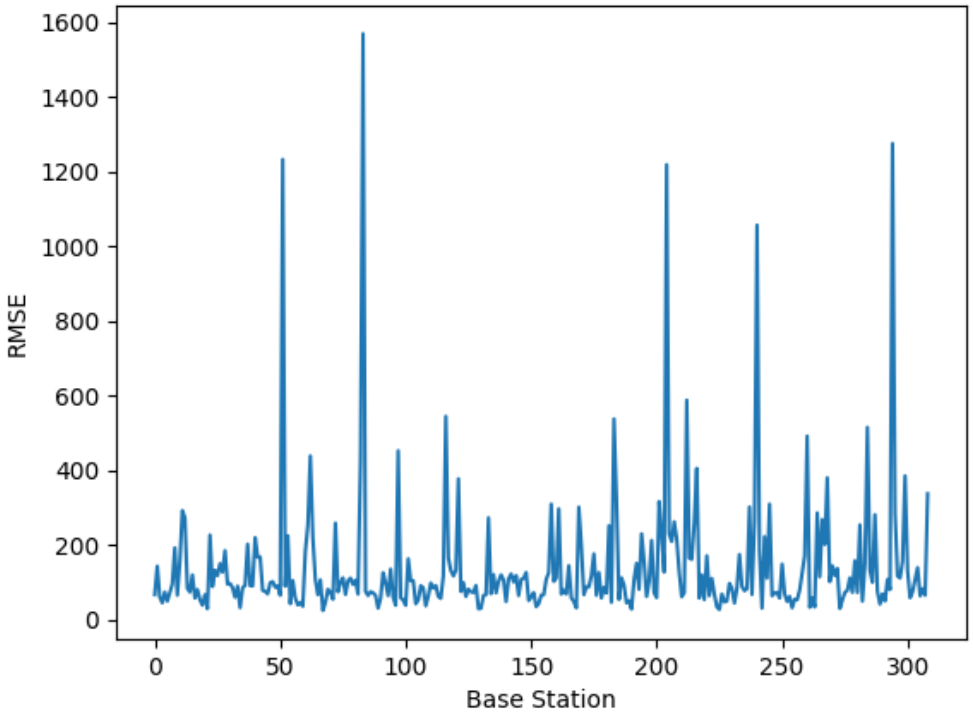

Figure 4.39 Mean squared ridge regression graph

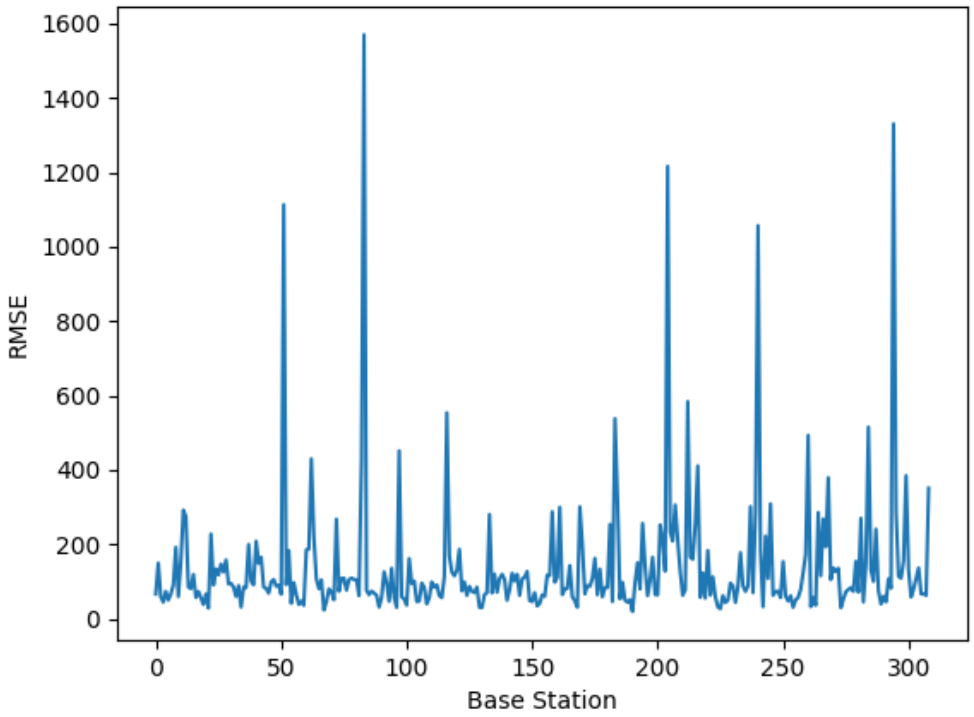

Figure 4.40 Mean squared error bayesian ridge regression graph

MSE shows us how close model fitted to the data points. It can be seen on the graphs on some of the machine learning algorithms results are great.

4.3.6 Root mean squared error (RMSE)

RMSE is used to measure the magnitude of error of a machine learning model, which is generally used to find the distance between the predicted values and original values. The RMSE is the standard deviation of the estimation errors. Errors are a measure of how far the regression line is from the data points. (Reyes, et al., 2010) The formula of RMSE can be seen on 4.16. where y is the observed value and $\hat{y}$ is the forecasts. Results of our algorithms can be seen in below histograms.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \quad (4.16)$$

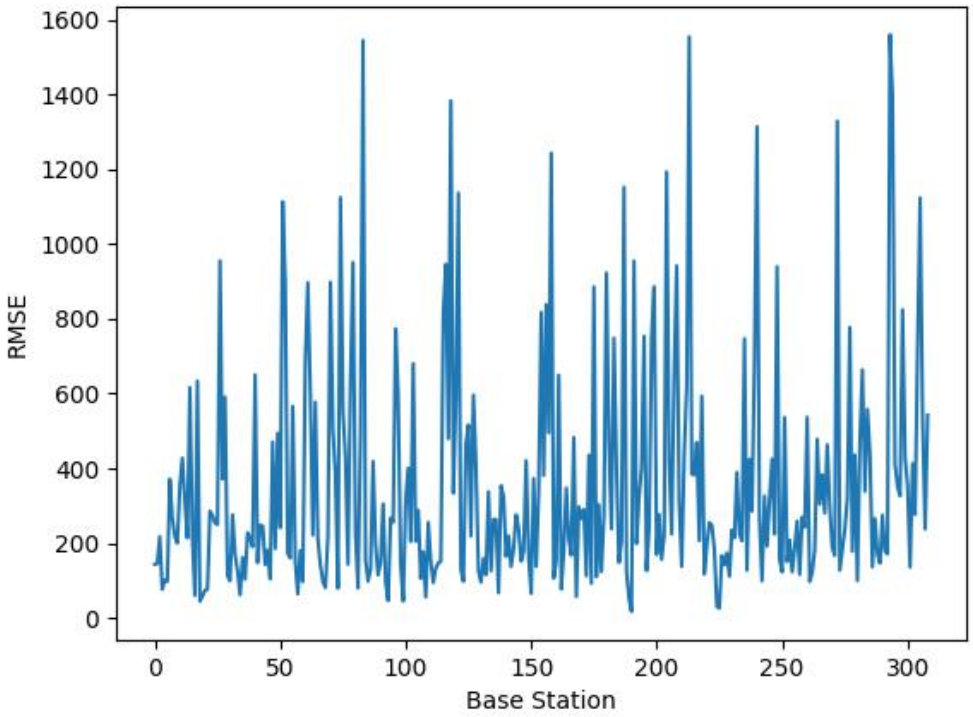

Figure 4.41 Root mean squared error gaussian processor graph

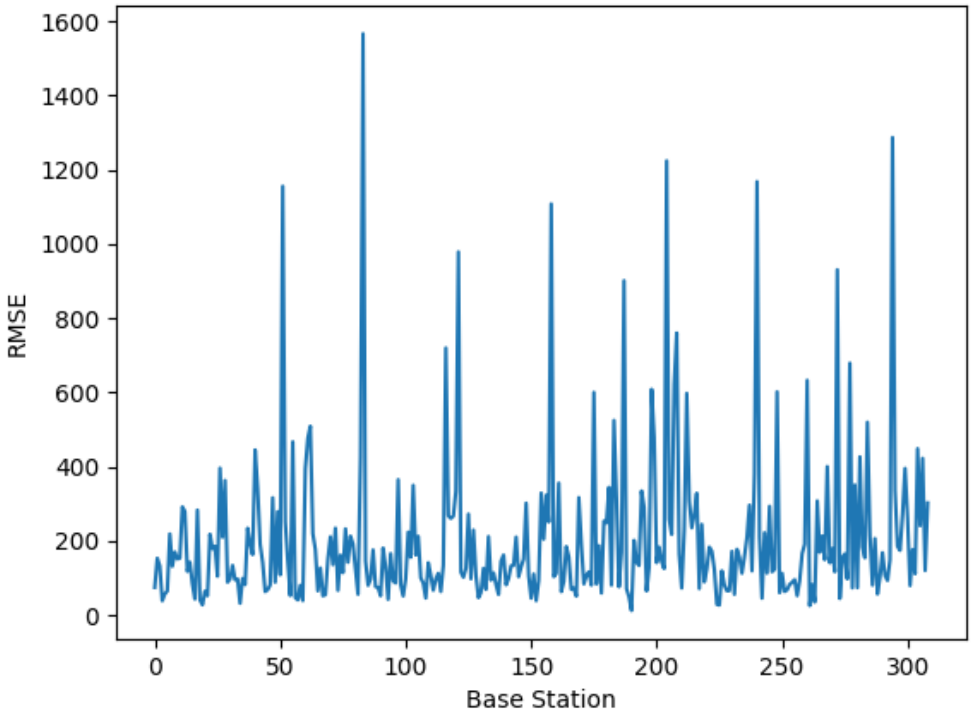

Figure 4.42 Root mean squared error kneighbor regressor graph

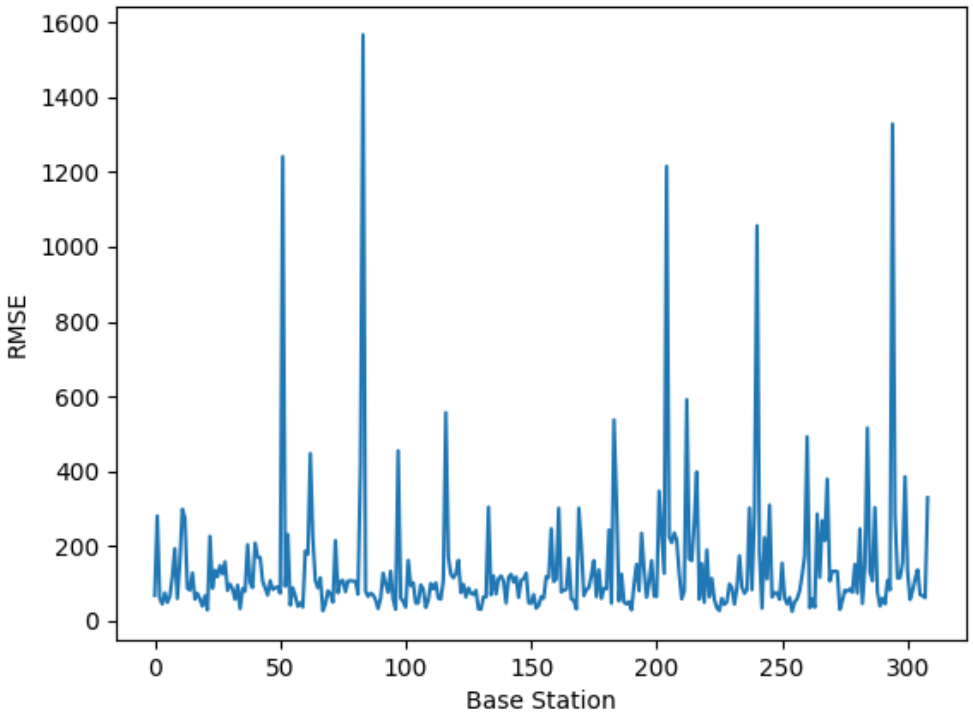

Figure 4.43 Root mean squared error linear regression graph

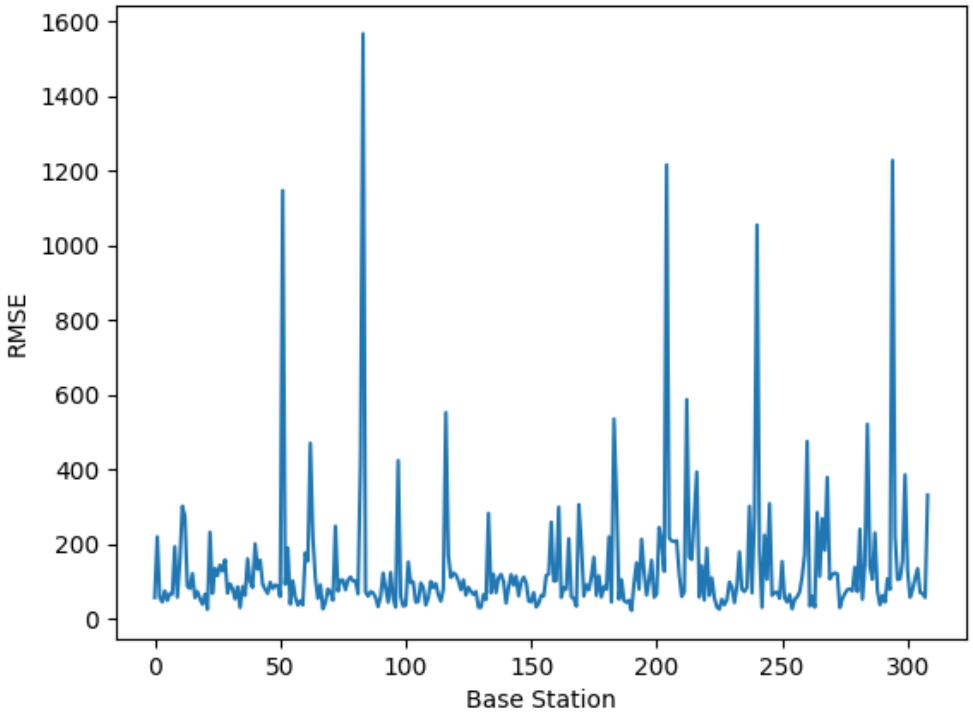

Figure 4.44 Root mean squared error ard regression graph

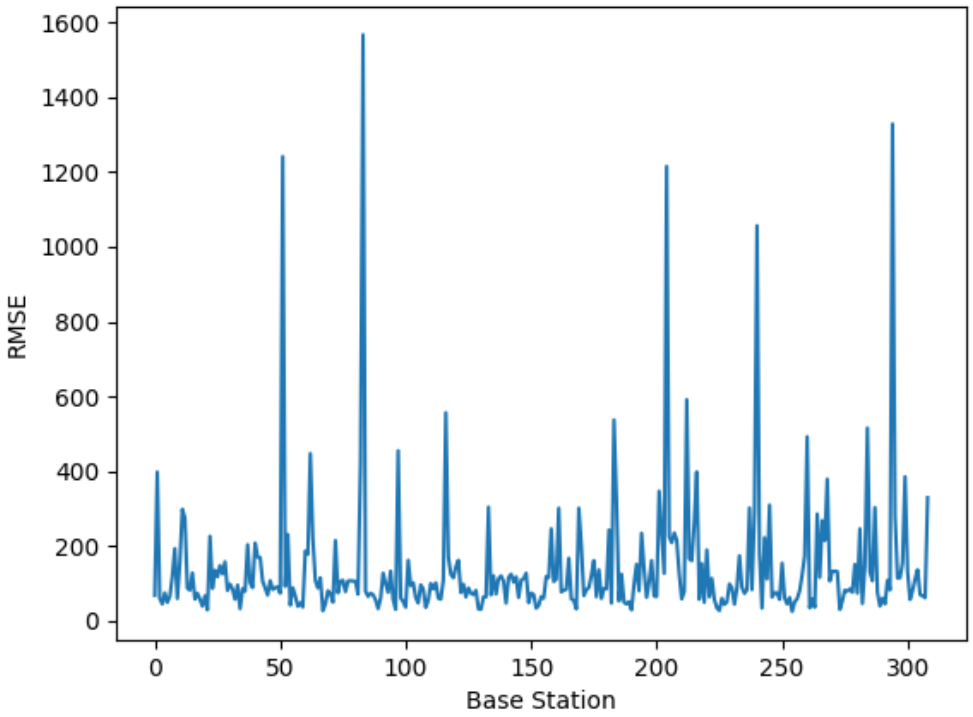

Figure 4.45 Root mean squared error lars regression graph

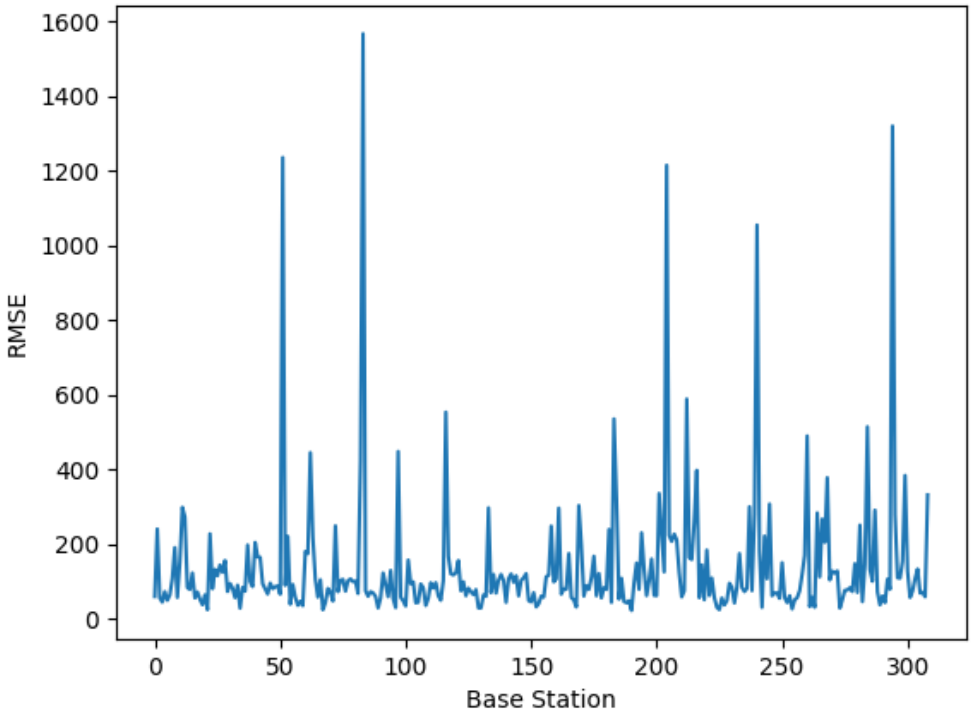

Figure 4.46 Root mean squared error lasso regression graph

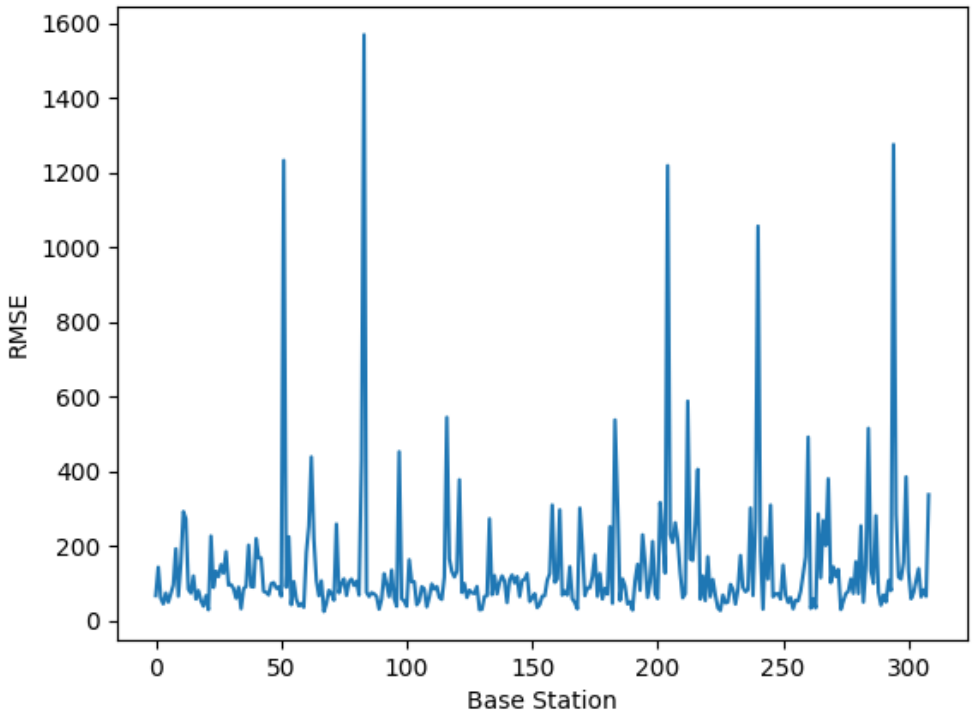

Figure 4.47 Root mean squared error ridge regression graph

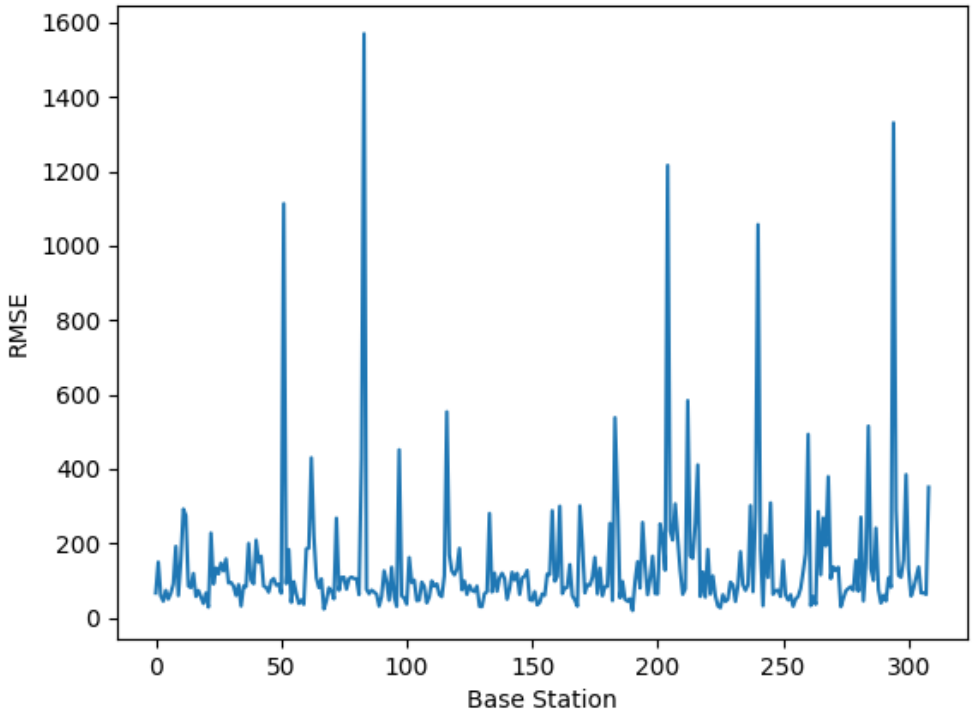

Figure 4.48 Root mean squared error bayesian ridge regression graph

4.4 Histogram of the Algorithms

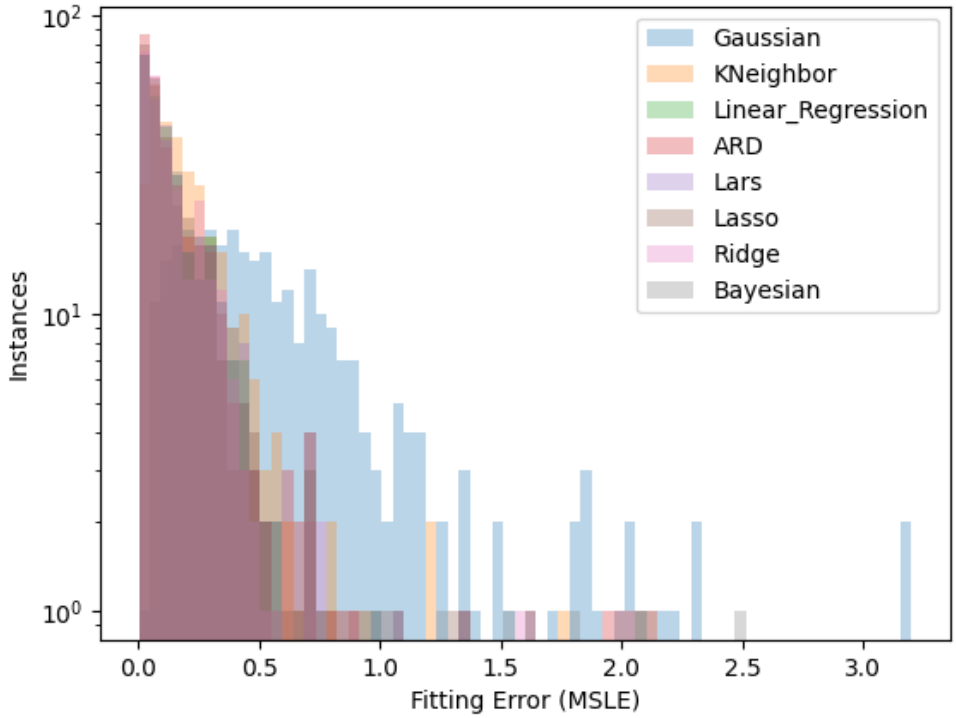

Figure 4.49 Mean squared logarithmic error histogram

We gathered the Mean Square Logarithmic Error values of all the algorithms we mentioned before and used their logarithm values in figure 1. Fitting error can be seen on the figure. Closer to the 0 smaller the error. Thus, Gaussian algorithm relatively performed poorly on some BTS.

5. CONCLUSION AND IMPLICATIONS

In this study, we gathered data from the field as TA as meters, RSRP as db and RSRQ as dbm with corresponding GPS coordinates as longitude and longitude. There was no additional hardware required for this study, we used data available on the BTS.

Before using this data, we performed data cleaning. We removed some of the BTS which had outlier values was affecting the result of the machine learning algorithms. The machine learning algorithms are run for all 366 distinct BTS.

As we mentioned earlier, for better accuracy, the geographic location of the BTS should be considered. To address that, we used different machine learning algorithms such as Gaussian Processor, KNeighbor Regressor, Linear Regression, ARD, Lars, Lasso, Ridge Regression and Bayesian Ridge Regression. The result of these machine learning algorithms compared to the GPS coordinates to measure our accuracy.

We evaluated the results of these machine learning algorithms using Mean Absolute Error, Mean Squared Error, Mean Squared Logarithmic Error, and R2.

The Mean Squared Logarithmic Error table can be seen in Table 1. We took logs to get rid of exponential expressions. Since some of the numbers are very large and some are very small numbers therefore log scale is preferred.

Most of the machine learning algorithms performed similarly on the calculation of the distance. We may include the BTS environment data to improve our results and make more precise results in our feature work.

Table 5.1 Mean squared logarithmic error results table

Machine Learning Algorithm	Mean Squared Log Error
Gaussian	0.59
KNeighbors	0.22
Linear Regression	0.17
Ard Regression	0.17
Lars	0.17
Lasso	0.16
Ridge	0.16
Bayes	0.19

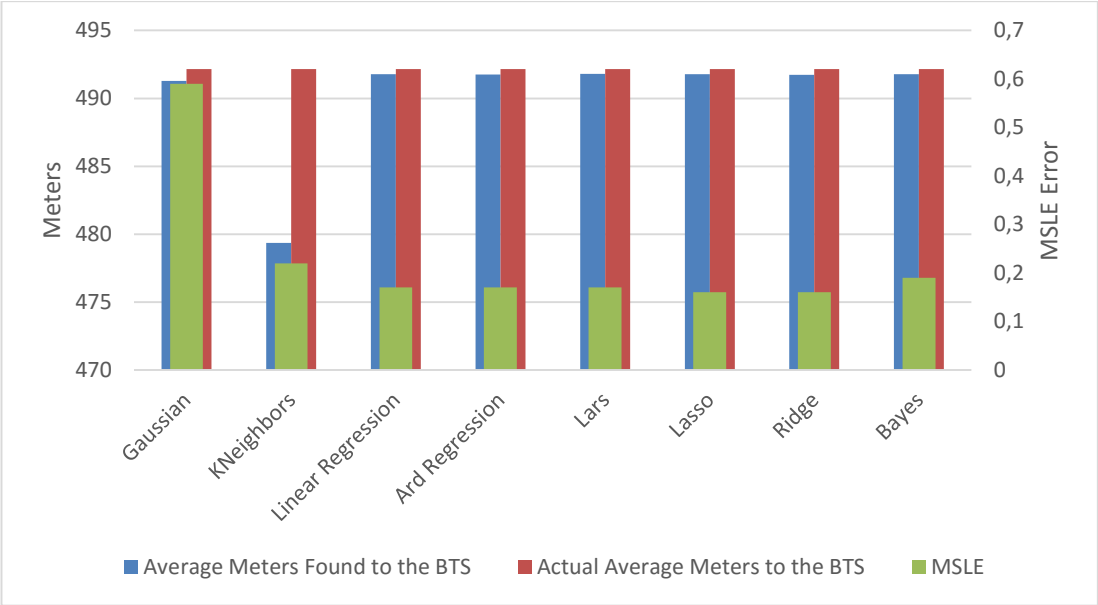

Figure 5.1 Actual distance to the BTS compared to the found meters by algorithms

REFERENCES

Van G. T., Suykens, J.A.K., De Moor, B., Vandewalle, J., 2001. Automatic Relevance Determination for Least Squares Support Vector Machine Regression, International Joint Conference on Neural Networks. Proceedings (Cat. No.01CH37222), Washington, 15-19 July 2001, 2416-2421.

Reyes, C., Hilaire, T., Paul, S., Mecklenbräuker, C. F., 2010. Evaluation of the Root Mean Square Error Performance of the PAST-Consensus Algorithm, 2010 International ITG Workshop on Smart Antennas (WSA), Bremen, 23-24 February 2010, 156-160.

Efron, B., Hastie, T., Johnstone, I., Tibshirani, R., 2004. Least Angle Regression The Annals of Statistics, Stanford, April 2004, 32(2), 407–499.

Pirzadeh, H., Wang, C., Papadopoulos, H., 2019. Machine-Learning Assisted Outdoor Localization via Sector-Based Fog Massive MIMO, 2019 IEEE International Conference on Communications (ICC), 20-24 May 2019, 1-6.

Anisetti, M., Ardagna, C. A., Bellandi, V., Damiani, E., Reale, S., 2011. Map Based Location and Tracking in Multipath Outdoor Mobile Networks, IEEE Transactions on Wireless Communications, 20 January 2011, 10(3), 814-824.

Samarah, K. G., 2016. Mobile Positioning Technique Based on Timing Advance and Microcell Zone Concept for GSM Systems, International Journal on Communications Antenna and Propagation (IRECAP), August 2016, 6(4), 211.

Ruan, W., Milstein, A. B., Blackwell, W., Miller, E. L., 2017. Multiple Output Gaussian Process Regression Algorithm for Multi-Frequency Scattered Data Interpolation, 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Fort Worth, 23-28 July 2017, 3992-3995.

Hirose, H., Soejima, Y., Hirose, K., 2012. NNRMLR: A Combined Method of Nearest Neighbor Regression and Multiple Linear Regression, 2012 IIAI International Conference on Advanced Applied Informatics, Fukuoka, 20-22 September. 2012, 351-356.

Lee, K., Lee, H., You, K., 2019. Optimised Solution for Hybrid TDOA/AOA-Based Geolocation Using Nelder-Mead Simplex Method, IET Radar, Sonar & Navigation, Volume 13, Issue 6, 992 – 997.

Martínez Hernández, L. A., Pérez Arteaga, S., Sánchez Pérez, G., Sandoval Orozco, L., García Villalba, L. J., 2019. Outdoor Location of Mobile Devices Using Trilateration Algorithms for Emergency Services, IEEE Access, Volume 7, 52052-52059,

Li, C., Li, W., 2010. Partial Least Squares Method Based on Least Absolute Shrinkage and Selection Operator. 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), Chengdu, 20-22 Aug. 2010 V4-591-V4-593.

Li, D., Ge, Q., Zhang, P., Xing, Y., Yang, Z., Nai, W., 2020. Ridge Regression with High Order Truncated Gradient Descent Method. 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), Hangzhou, 2020, 22-23 Aug. 2020, 252-255.

Pereira, R. C., Abreu, P. H., Rodrigues, P. P., 2020. VAE-BRIDGE: Variational Autoencoder Filter for Bayesian Ridge Imputation of Missing Data, 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, 19-24 July 2020, 1-7.

Divyanshu, M. 2021. Regression: An Explanation of Regression Metrics And What Can Go Wrong, 07/06/2021. https://towardsdatascience.com/regression-an-explanation-of-regression-metrics-and-what-can-go-wrong-a39a9793d914

Rosenthal, J., 2011. Statistics and Data Interpretation for Social Work, 512, Springer Publishing Company

APPENDICES

Appendix A. Source Code

Appendix A. Source Code

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn import metrics
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import ARDRegression, LinearRegression, BayesianRidge, Lasso, La
rs
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import DotProduct, WhiteKernel, RBF, ConstantKernel
from sklearn.neighbors import KNeighborsRegressor
from sklearn import linear_model
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import load_digits


def regression_results(y_true, y_pred):

    # Regression metrics
    explained_variance=metrics.explained_variance_score(y_true, y_pred)
    mean_absolute_error=metrics.mean_absolute_error(y_true, y_pred)
    mse=metrics.mean_squared_error(y_true, y_pred)

    mean_squared_log_error = "nan"
    try:
        mean_squared_log_error=metrics.mean_squared_log_error(y_true, y_pred)
        mean_squared_log_error = round(mean_squared_log_error,4)
    except:
        pass
    median_absolute_error=metrics.median_absolute_error(y_true, y_pred)
    r2=metrics.r2_score(y_true, y_pred)

    res = {}
    res['explained_variance'] = round(explained_variance,4)
    res['mean_squared_log_error'] = mean_squared_log_error
    res['r2'] = round(r2,4)
    res['MAE'] = round(mean_absolute_error,4)
    res['MSE'] = round(mse,4)
    return res
```

```
def process_ard(X_train, X_test, y_train, y_test):

    model = ARDRegression(threshold_lambda=1e5)
    model.fit(X_train, y_train)

    return model

def process_bayes(X_train, X_test, y_train, y_test):


    model = BayesianRidge()

    model.fit(X_train, y_train)

    return model

def process_gaussian(X_train, X_test, y_train, y_test):


    kernel = ConstantKernel(0.1, (0.01, 10.0))
    model =  GaussianProcessRegressor(kernel=kernel, random_state=None,  n_restarts_optimizer
=10, normalize_y=False, optimizer='fmin_l_bfgs_b').fit(X, y)
    model.fit(X_train, y_train)
    return model

def process_kneighbor(X_train, X_test, y_train, y_test):

    model = KNeighborsRegressor(n_neighbors=8)
    model.fit(X_train, y_train)

    return model

def process_lars(X_train, X_test, y_train, y_test):

    model = Lars()
    model.fit(X_train, y_train)

    return model


def process_lasso(X_train, X_test, y_train, y_test):

    model = Lasso(alpha=1.0, max_iter=50000)
    model.fit(X_train, y_train)

    return model
```

```
def process_linearRegression(X_train, X_test, y_train, y_test):

    model = linear_model.LinearRegression()
    model.fit(X_train, y_train)

    return model

def process_Ridge(X_train, X_test, y_train, y_test):


    model = linear_model.Ridge(alpha=.5)
    model.fit(X_train, y_train)

    return model



algos = [[process_ard, "ARD", "results_ard.csv"],
[process_bayes, "Bayes_Ridge", "results_bayes.csv"],
[process_gaussian, "Gaussian", "results_gaussian.csv"],
[process_kneighbor, "KNeighbor", "results_kneighbor.csv"],
[process_lars, "Lars", "results_lars.csv"],
[process_lasso, "Lasso", "results_lasso.csv"],
[process_linearRegression, "Linear_Regression", "results_linear_regression.csv"],
[process_Ridge, "Ridge", "results_ridge.csv"]]


# prepare results
for algo in algos:
    f = open(algo[2],'w')
    f.write('id\tmean_squared_log_error\texplained_variance\tr2\tMAE\tMSE\n')
    f.close()


all_data = pd.read_csv('lat_lon.csv')

all_data = all_data[all_data['distance'] < 5000]

groups = all_data.groupby(['cellid'])

results = pd.DataFrame()

for tmp in groups:

    cell_df = tmp[1]

    cell_df['last_rsrq'] = cell_df['last_rsrq'].astype('int')
    cell_df['distance'] = cell_df['distance'].astype('float')
```

```
    cell_df['fist_ta'] = ((cell_df['fist_ta']) / 16 * 78 + 39).astype('int')
    cell_df['last_ta'] = ((cell_df['last_ta']) / 16 * 78 + 39).astype('int')


    feature_df = cell_df[['fist_ta', 'last_ta',  'last_rsrp', 'last_rsrq', 'lastutsirn','lastcqi']]

    X = np.asarray(feature_df)
    y = np.asarray(cell_df['distance'])

    if len(cell_df.index) <= 20:
        print("not enough data for ", tmp[0], len(cell_df.index))
        continue


    print('processing %s with %d samples' % (tmp[0], len(cell_df.index)))

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=4)
    scaling = MinMaxScaler(feature_range=(-1,1)).fit(X_train)
    X_train = scaling.transform(X_train)
    X_test = scaling.transform(X_test)

    for algo in algos:
        print ("\t running: ", algo[1])

        model = algo[0](X_train, X_test, y_train, y_test)
        y_predict = model.predict(X_test)

        res = regression_results(y_test, y_predict)
        f = open(algo[2],'a')

        s = '%d\t%s\t%f\t%f\t%f\t%f\t%f\n'%(tmp[0], str(res['mean_squared_log_error']), res['expla
ined_variance'], res['r2'], res['MAE'], res['MSE'], res['RMSE'])
        f.write(s)
        f.close()

        cell_df[algo[1]] = model.predict(scaling.transform(X))

    results = pd.concat([results, cell_df])


results.to_csv(path_or_buf= "results.csv",header=True, mode="w")
```

BIBLIOGRAPHY

Name Surname : Ahmed Hakan KILIÇ

Education

Bachelor's Degree : Istanbul Arel University, Faculty of Engineering, Department of Computer Engineering - 2016

Postgraduate Degree : Istanbul Commerce University, Graduate School of Natural and Applied Sciences, Department of Computer Engineering - 2021

Publications

Kılıç A. H., Boyacı, A., 2021. Location Estimation on Mobile Networks. Istanbul Commerce University Journal of Technologies and Applied Sciences, In Press.