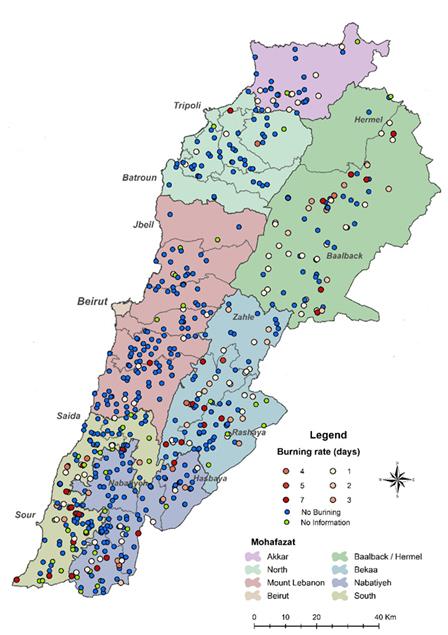
**Methodology:**

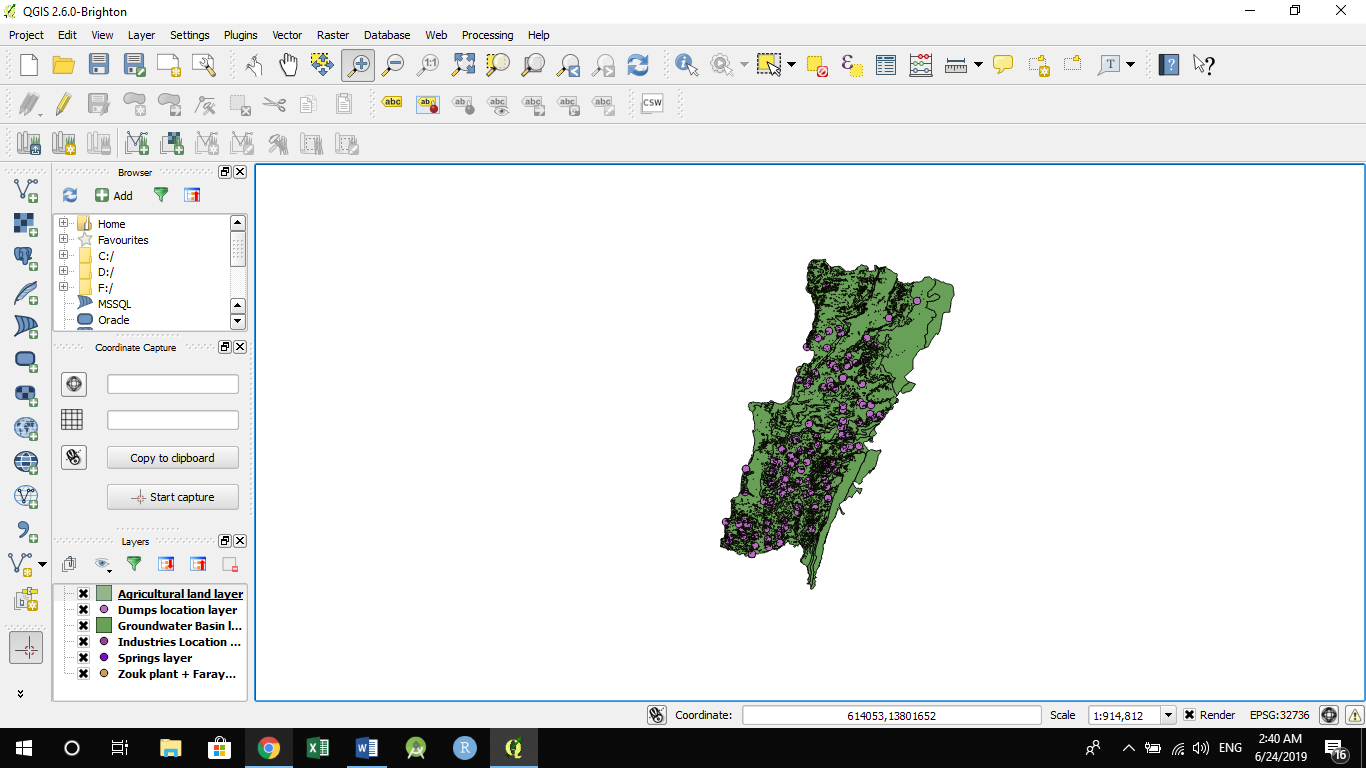
1. **Slope and Traffic data collection**:

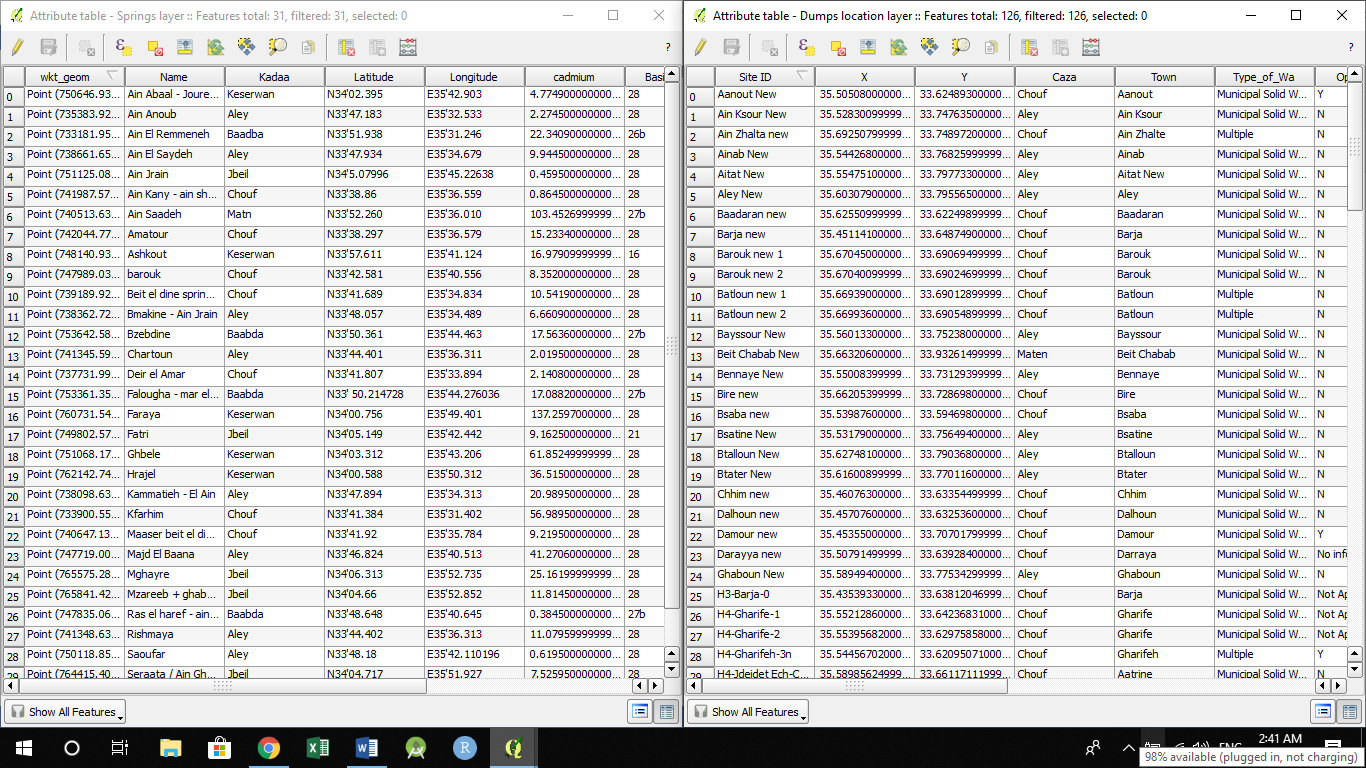
The slope values were obtained using the QGIS application, where first we identified where each spring is located on the map. Then we got the coordinates of each of these springs and the coordinates of the three closest dumps. Then, using the slope equation, we calculated the slope of each spring with these dumps and then got the average of the result.

For the traffic part, we used the Google API which provides a consistent mapping service, where it gives us the directions, the traffic, and the road. It provides real-time traffic and an average of the traffic over the year. We had to use the distance matrix which gives us the travel time and the distance between the dumps and each of the springs and the nearest routes to the springs which show how close the emission of the car is. Looking into to the below graph: the values were put according to the intensity of the traffic, where green means no traffic at all which was given a value of 0, yellow means very little amount of cars are traveling which we gave a value of 1, red means there's a lot of traffic and was given a value of 2, and the maroon color was given a value of 3 which meant extreme traffic on a daily basis.



1. **Slope Calculation:**

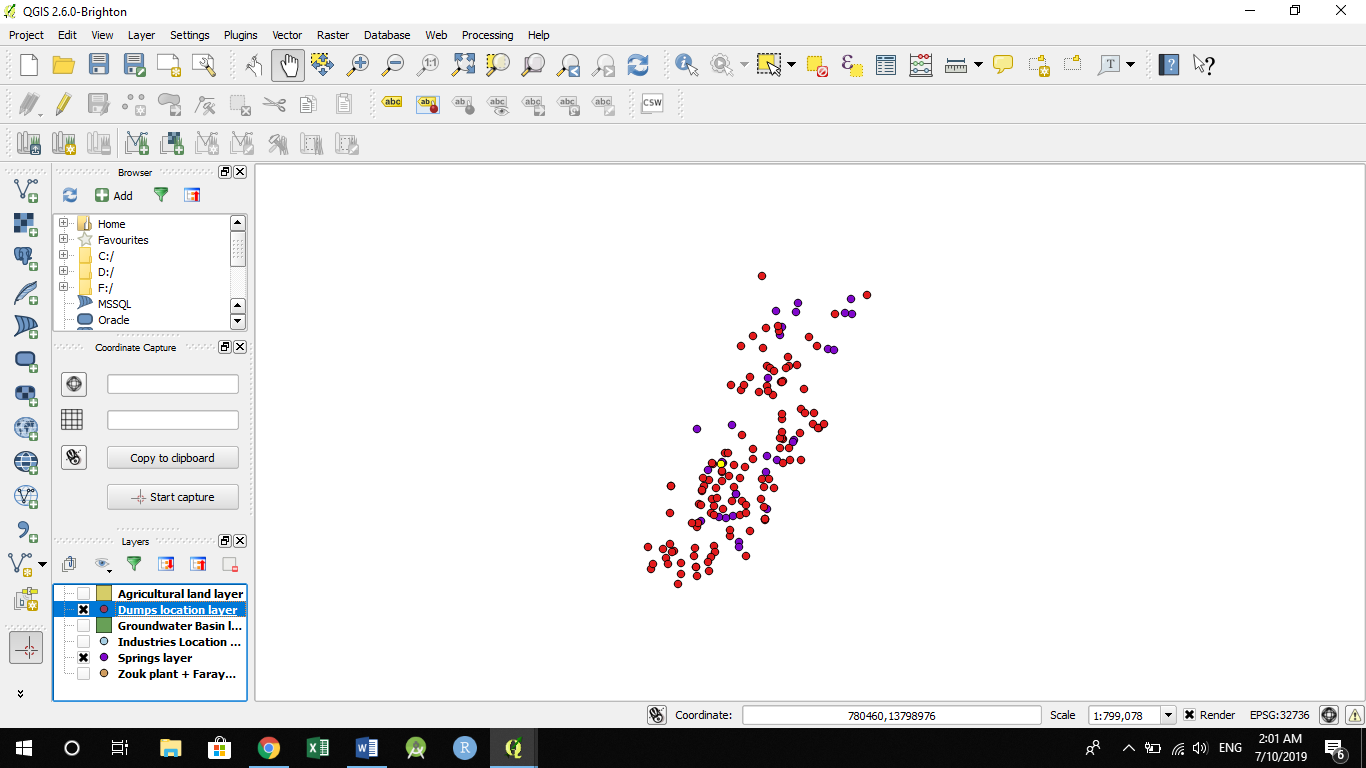




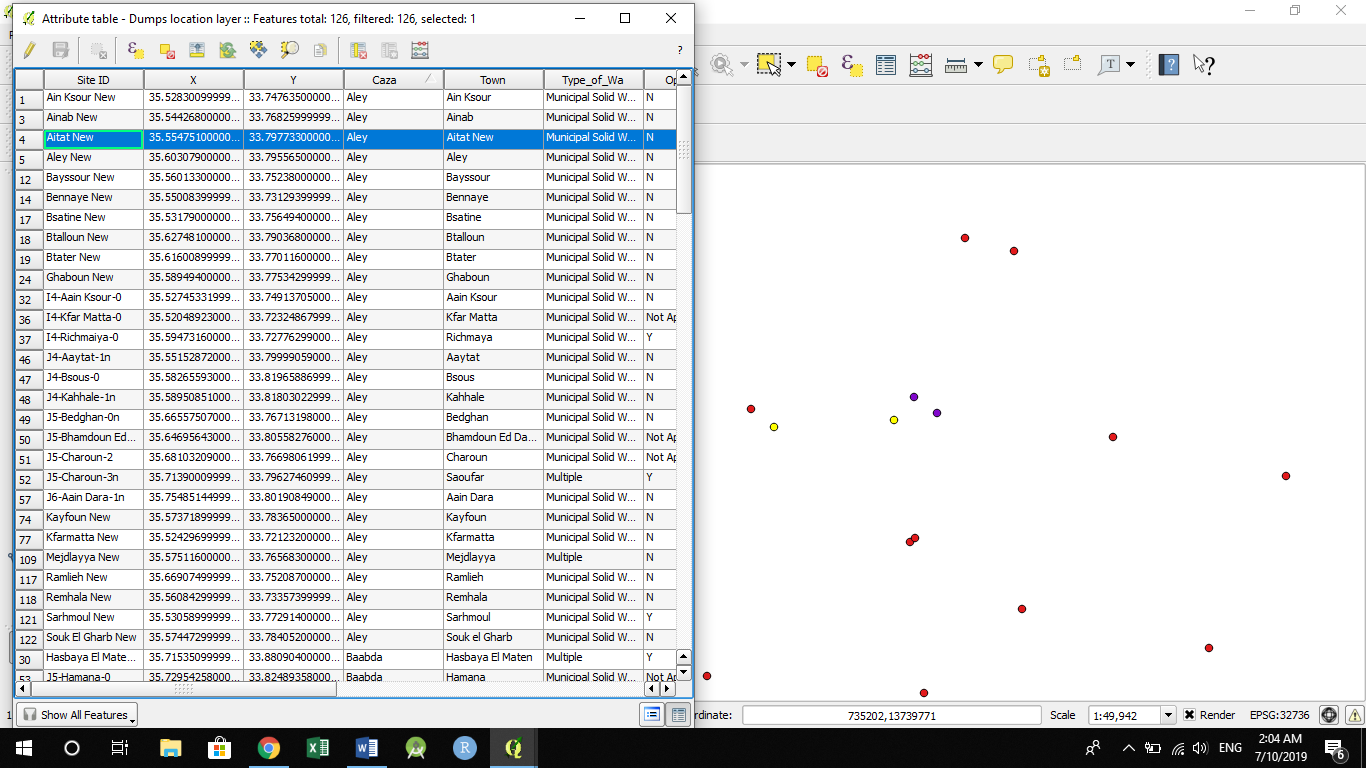
First we choose one from the spring layer table which will turn one circle into yellow, then we searched for the closest in dumps location layer table. Then we used the x and y of each of them to calculate the slope using this equation (y2 - y1) / (x2 - x1).

Example:

After choosing Kammatieh - El Ain from kadaa aley we get x1= 33.894 and y1=35.313



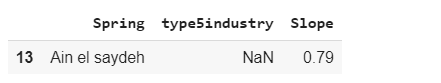
After zooming in we got the closest dump which is aitat new where we got x2=35.554751 y2=33.797733



Which gives us the slope of -0.91

**Results**

1. **EDA:**
2. The data has 31 instances (Springs) with 69 features. Main features:
   * type5industry: Presence of type 5 manufacturing of jewelry and fine stone industries
   * Slope: Slope between the closest dump and the spring
   * Traffic: Amount of Traffic per district on a scale of 0 to 3 (low to high)
   * type8industry: Presence of type 8 paper and paper product indutries
   * Max\_distdump: Maximum distance between spring and industry
   * Precipitation: The amount of rain per year on each spring in mm
   * Mean\_yrs: Mean operating years of the dumps
   * type14agri: Presence of type 14 "citrus trees" agricultural lands
   * type3agri: Presence of type 3 "banana trees" agricultural lands
   * type7agri: Presence of type 7 "greenhouses" agricultural lands
   * For the remaining features please refer to the legend sheet.
3. We loaded the excel sheet into Panda data frame. Then we checked for entries that have empty value(NAN value) and we found one instance with NAN value for type5industry feature:



For type5industry feature with NaN value, we replaced it with (0, which means there is no present for type 5 industry in this region).

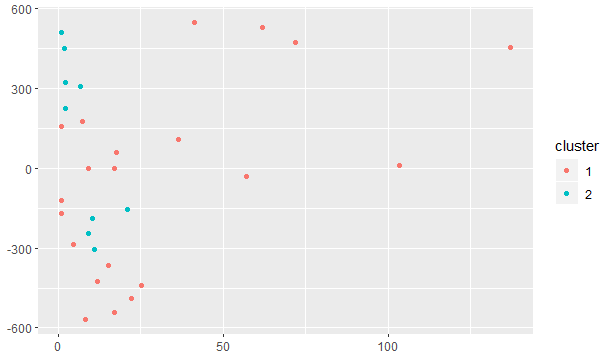
1. We calculated the correlation between each feature and Cadmium concentration. The top correlated features with CD concentration:

Traffic(0.80), type8industry(-0.341), Max\_distdump(-0.338), Precipitation(0.330),While the least correlated features with CD concentration: Mean\_yrs(0.085), type14agri(0.061), type3agri(0.086), type7agri(-0.059)

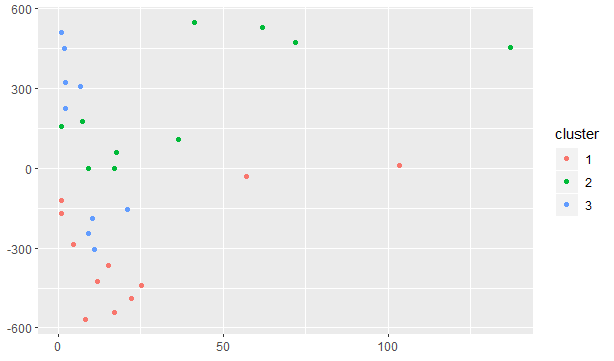
1. We scaled all the features value to be between (-1,1) to make sure that big values don’t overwhelm small values. I used the min-max formula to do the normalization.(For visualization and plots, I used the original values of features).
2. Then we did K-means clustering by including only the top correlated features with CD concentration (Traffic, type8industry, Max\_distdump, Precipitation). So we processed by choosing K equal to 2 and 3.

The Plots for each model are as follows:

For K = 2:



K = 3:



* 1. **Clusters summary:**

K = 2:

|  |  |  |
| --- | --- | --- |
| **Cluster\_mean\_value** | **Cluster 1:23** | **Cluster 2:7** |
| **Cd\_con** | 23.76138 | 26.79814 |
| **Traffic** | 0.8695652 | 1 |
| **type8industry** | 2.9565217 | 0.5714286 |
| **Max\_distdump** | 27271.17 | 15242.66 |
| **Precipitation** | 1226.087 | 1328.571 |

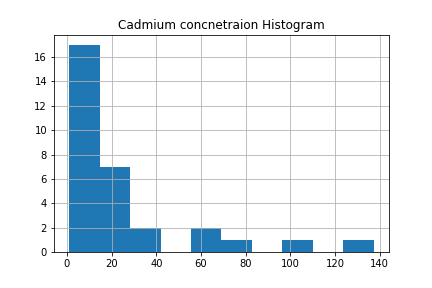
K = 3:

|  |  |  |  |
| --- | --- | --- | --- |
| **Cluster\_mean\_value** | **Cluster 1:9** | **Cluster 2:7** | **Cluster 3:14** |
| **Cd\_con** | 39.74891 | 26.79814 | 13.48369 |
| **Traffic** | 1.1111111 | 1 | 0.7142857 |
| **type8industry** | 0.0000000 | 0.5714286 | 4.8571429 |
| **Max\_distdump** | 10435.91 | 15242.66 | 38093.84 |
| **Precipitation** | 1266.667 | 1328.571 | 1200.000 |

We could conclude from the results that when K=3 the algorthim clusters the samples better than K=2 in term of Cd\_con. The CD\_con mean values for cluster 1 , 2 , and 3 are equal to 39.74891, 26.79814 , 13.48369 respectively. Which shows that the CD\_con is spread low, medium, and high in each cluster.

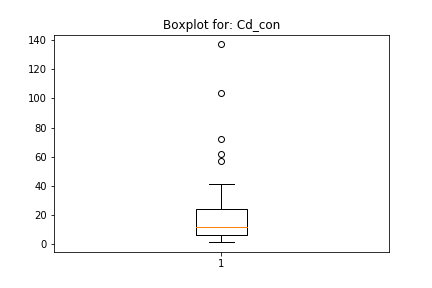
1. Since our objective is to predict the amount Cadmium concentration in the spring; thus, I plotted the histogram for Cadmium concentration and calculated the following statistics measures for Cadmium concentration:

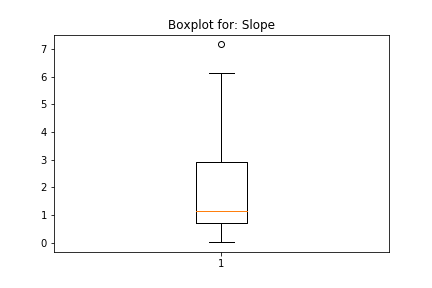
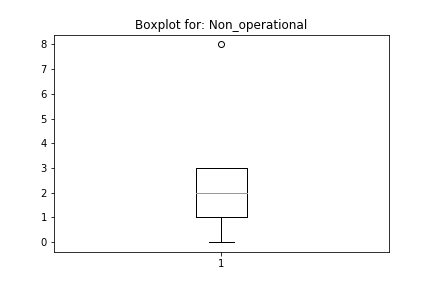
* count 31.000000
* mean 24.001398
* std 31.885968
* min 1.000000
* 25% 5.717900
* 50% 11.079600
* 75% 23.751450
* max 137.259700

****

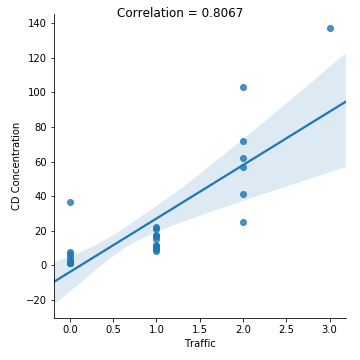
From the measurements and the histogram, we conclude that the distribution data of Cadmium concentration is not a normal distribution.

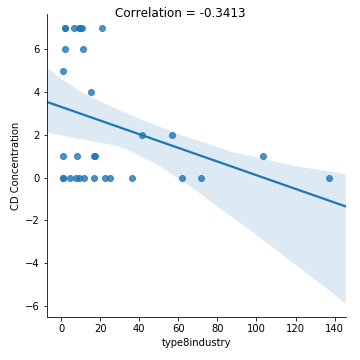
1. We plotted the Boxplot for each feature. For CD concentration, we found that there are 5 points that are outliers (all points with CD concentration >40). Due to small data size, we decided to not exclude these outliers.

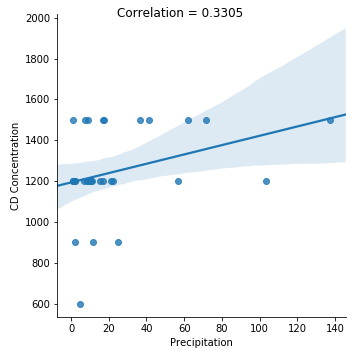
****

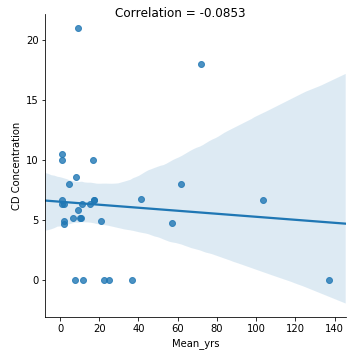
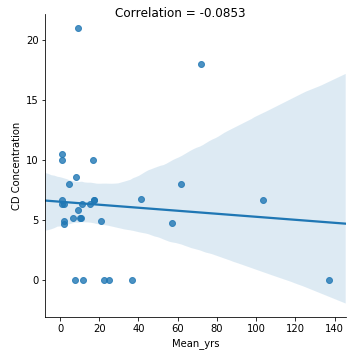
****

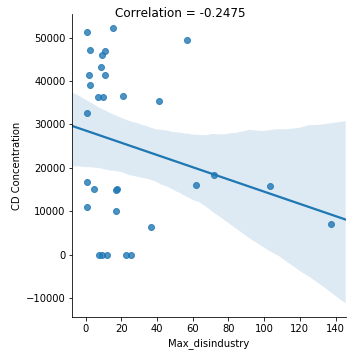
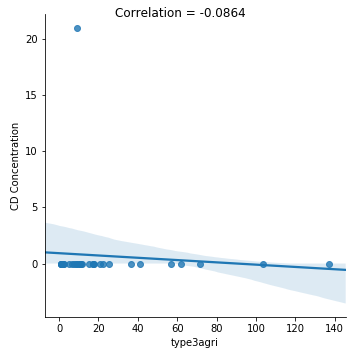
1. We plotted the correlation between each feature and Cadmium concentration. Below are the most correlated feature plots:

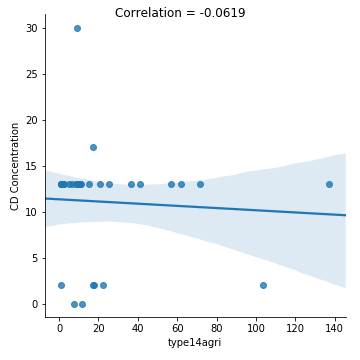
****

****

****

****

****

****

1. **Feature selection & Modeling:**

We had conducted feature selection using four different methods:

1-**Decision trees** top 10 features: 'Traffic','Slope','Precipitation','Min\_distagri','SlopeInd','Mean\_volume','Max\_distdump','Max\_area', 'Max\_yrs','Median\_distagri'

2- **PCA** top 10 features:

'Type2industry','Max\_disindustry', 'Max\_distdump', 'Total\_industry','Multipledumps', 'type5agri', 'type15agri', 'type7industry', 'type3industry', 'Mean\_volume'

3- **Univariate** top 10 features:

'Traffic','Mean\_distdump','Median\_distdump','Max\_distdump','Precipitation', 'Median\_distagri','Max\_yrs', 'type3agri', 'Mean\_area' ,'type5industry'

4-**Recursive feature Elimination** top 10 features: 'Mean\_distdump','Traffic','type15agri','type5agri','Min\_distagri',type4industry', 'Min\_areaagri', 'type3industry', 'type8industry','type9industry'

Then we proceeded with modeling by taking the set of three top features from each method listed above; thus, we ended up with the following features: **'Slope','Max\_disindustry','Traffic','Type2industry','Mean\_distdump'**

As for modeling, we started by tuning the hyper-parameters using leave one out cross validation for each of the following model:

**1- SVM with polynomial kernel**

**2- SVM with RBF kernel**

**3- SVM with linear kernel**

**4- M5 regression trees**

**5- Random Forest**

**6- GradientBoostingRegressor (GBDT )**

**7- CatBoostRegressor**

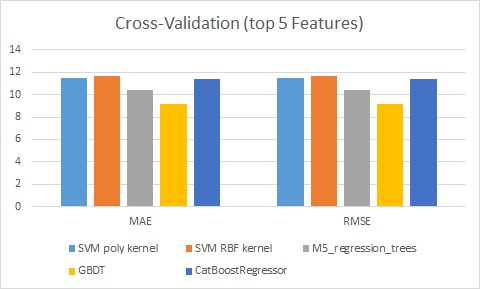
Note that the optimal hyper-parameters are chosen based on the **RMSE** score, which is equivalent to the **MSE** score in the case of leave one out cross validation.

After obtaining the top hyper-parameters for each of the model above, we re-trained each model on the training data with the optimal hyper-parameters. Finally, we let the predictors to forecast on the testing dataset. Find the experiment results below in the results section

**2.1) Results:**

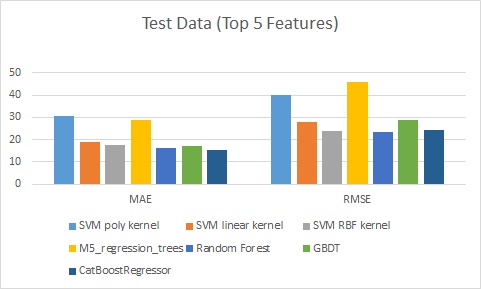
**Cross validation results for the top selected features (5 features):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model / Metric | MSE | MAE | RMSE | CAD MEAN | Correlation: y\_true & y\_preds |
| SVM poly kernel | 327.96 | 11.46 | 11.46 | 21.37 | 0.84 |
| SVM linear kernel | 488.38 | 12.58 | 12.58 | 0.73 |
| SVM RBF kernel | 298.02 | 11.66 | 11.66 | 0.82 |
| M5\_regression\_trees | 352.06 | 10.45 | 10.45 | 0.78 |
| Random Forest | 353.42 | **9.54** | **9.54** | **0.80** |
| GBDT | 290.84 | **9.13** | **9.13** | **0.82** |
| CatBoostRegressor | 654.96 | 11.40 | 11.40 | 0.57 |

****

**Test data results for the top selected features(5 features):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model / Metric | MSE | MAE | RMSE | MAPE | R2 | CAD MEAN | Correlation: y\_true & y\_preds |
| SVM poly kernel | 1601.89 | 30.54 | 40.02 | 150.70 | -0.40 | 21.37 | 0.40 |
| SVM linear kernel | 787.64 | 18.88 | 28.06 | 59.44 | 0.31 | 0.68 |
| SVM RBF kernel | 574.92 | 17.41 | 23.97 | 60.89 | 0.49 | 0.715 |
| M5\_regression\_trees | 2093.73 | 28.96 | 45.75 | 152.92 | -0.83 | 0.42 |
| Random Forest | 553.28 | 16.18 | 23.52 | 76.66 | 0.51 | 0.72 |
| GBDT | 817.28 | 17.32 | 28.58 | 75.07 | 0.28 | 0.65 |
| **CatBoostRegressor** | 598.26 | **15.39** | **24.45** | **36.71** | **0.47** | **0.78** |

****

**Cross validation results for the all features (66 features):**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model / Metric | MSE | MAE | RMSE | CAD MEAN | Correlation: y\_true & y\_preds |
| SVM poly kernel | 867.91 | 16.75 | 16.75 | 21.37 | 0.39 |
| SVM linear kernel | 962.50 | 16.70 | 16.70 | 0.16 |
| SVM RBF kernel | 927.34 | 16.53 | 16.53 | 0.31 |
| **M5\_regression\_trees** | **397.14** | **9.71** | **9.71** | **0.81** |
| Random Forest | 442.01 | 11.20 | 11.20 | 0.73 |
| GBDT | 444.04 | 11.16 | 11.16 | 0.73 |
| CatBoostRegressor | 426.43 | 11.44 | 11.44 | 0.75 |

**Test data results for the all features (66 features):**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Model / Metric | MSE | MAE | RMSE | MAPE | R2 | CAD MEAN | Correlation: y\_true & y\_preds |
| SVM poly kernel | 1530.99 | 24.42 | 39.12 | 50.02 | -0.34 | 21.37 | 0.09 |
| SVM linear kernel | 1442.05 | 23.11 | 37.97 | 59.85 | -0.26 | 0.28 |
| SVM RBF kernel | 1457.42 | 22.38 | 38.17 | 49.14 | -0.27 | 0.34 |
| **M5\_regression\_trees** | **323.69** | **14.27** | **17.99** | **90.79** | **0.71** | **0.86** |
| Random Forest | 1865.64 | 26.95 | 43.19 | 139.61 | -0.63 | 0.39 |
| GBDT | 1548.60 | 24.56 | 39.35 | 109.19 | -0.35 | 0.476 |
| CatBoostRegressor | 1730.55 | 27.04 | 41.59 | 93.80 | -0.51 | 0.21 |

We can conclude the following from the above results:

In the case of using the top 5 selected features, **CatBoostRegressor** outperformed other models with MSE, MAE, RMSE, MAPE, R2 , and correlation between y\_true and y\_preds equal to 15.4, 24.4, 36.7, 0.47, and 0.78; respectively.

In the case of using all the features (66 features), **M5\_regression\_trees** outperformed other models with MSE, MAE, RMSE, MAPE, R2, and correlation between y\_true and y\_preds equal to 323.7, 14.27, 17.99, 90.79, and 0.71; respectively.

In my point of view, **M5\_regression\_trees with the all features (66 features) is the perfect model to use as a final one.**

#Optimum hyper-parameters for the case of using all features (66 features):

1-SVM with Poly: {'C': 0.1, 'degree': 3, 'gamma': 0.1, 'kernel': 'poly'}

2-SVM with linear: {'C': 0.1, 'kernel': 'linear'}

3-SVM with RBF: {'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}

4-Random Forest: {'bootstrap': False, 'criterion': 'mae', 'n\_estimators': 150, 'random\_state': 30}

5-M5 regresson: {'criterion': 'mae', 'max\_features': 'auto', 'presort': True, 'random\_state': 30}

6-GBDT: {'criterion': 'friedman\_mse', 'learning\_rate': 0.1, 'loss': 'ls', 'max\_depth': 6, 'max\_features': 'auto', 'n\_estimators': 160, 'random\_state': 30}

**7- catboostregressor: {'depth': 3, 'l2\_leaf\_reg': 3, 'learning\_rate': 0.3, 'verbose': 0}**

#Optimum hyper-parameters for the case of using the top five selected features:

1-SVM with Poly: {'C': 0.1, 'degree': 3, 'gamma': 1, 'kernel': 'poly'}

2-SVM with linear: {'C': 1000, 'kernel': 'linear'}

3-SVM with RBF: {'C': 1000, 'gamma': 0.1, 'kernel': 'rbf'}

4-Random Forest: {'bootstrap': True, 'criterion': 'mse', 'n\_estimators': 200, 'random\_state': 30}

**5-M5 regresson: {'criterion': 'mae', 'max\_features': 'auto', 'presort': True, 'random\_state': 30}**

6-GBDT: {'criterion': 'friedman\_mse', 'learning\_rate': 0.1, 'loss': 'huber', 'max\_depth': 3, 'max\_features': 'auto', 'n\_estimators': 160, 'random\_state': 30}

7- catboostregressor: {'depth': 3, 'l2\_leaf\_reg': 10, 'learning\_rate': 0.01, 'verbose': 0}