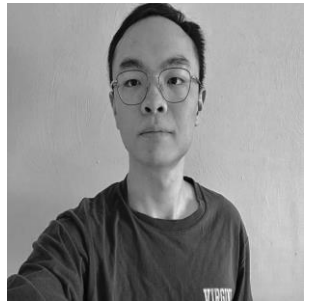
12-750 INFRASTRUCTURE MANAGEMENT PROJECT
SPRING 2022

Predicting Inspection Priority and Pipe Breaks in Water Distribution Network Using ML

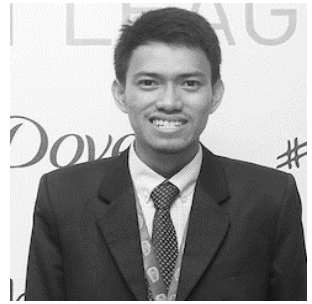
A Statistical 'Learning and Modeling' Study for Infrastructure Management

Devashri Karve
Graduate Student
dkarve@andrew.cmu.edu
CEE Department
Carnegie Mellon University

Tanay Kulkarni
Graduate Student
tskulkar@andrew.cmu.edu
CEE Department
Carnegie Mellon University



Yijie Zhu
Graduate Student
yijiezhu@andrew.cmu.edu
CEE Department
Carnegie Mellon University



Zulkifli Palinrungi
Graduate Student
zpalinru@andrew.cmu.edu
Energy Science,
Technology, and Policy
Carnegie Mellon University

Mentor
Prof. Dr. Donald Coffelt

Outline

- Aim and Scope
- Domain Introduction, GIS Maps, Data Exploration
- Methodology
- Clustering:
  - K-Means Clustering
- Inspection Priority Prediction:
  - Decision Tree Classification
  - Random Forest Classification
- Break Prediction:
  - Decision Tree Regression
  - Random Forest Regression
- Scores, Results, Outcomes, and Maps
- Conclusion

Aim and Objective

Aim:

Analyse water infrastructure pipe data to help utilities prioritize asset wise inspection and predict future pipe breaks based on historical data.

Objective:

- Using Machine Learning to group pipes with similar traits,
- predict inspection priorities based on geo-hydro-physical data, and
- predict number of breaks on individual pipes

Domain: Water Infrastructure

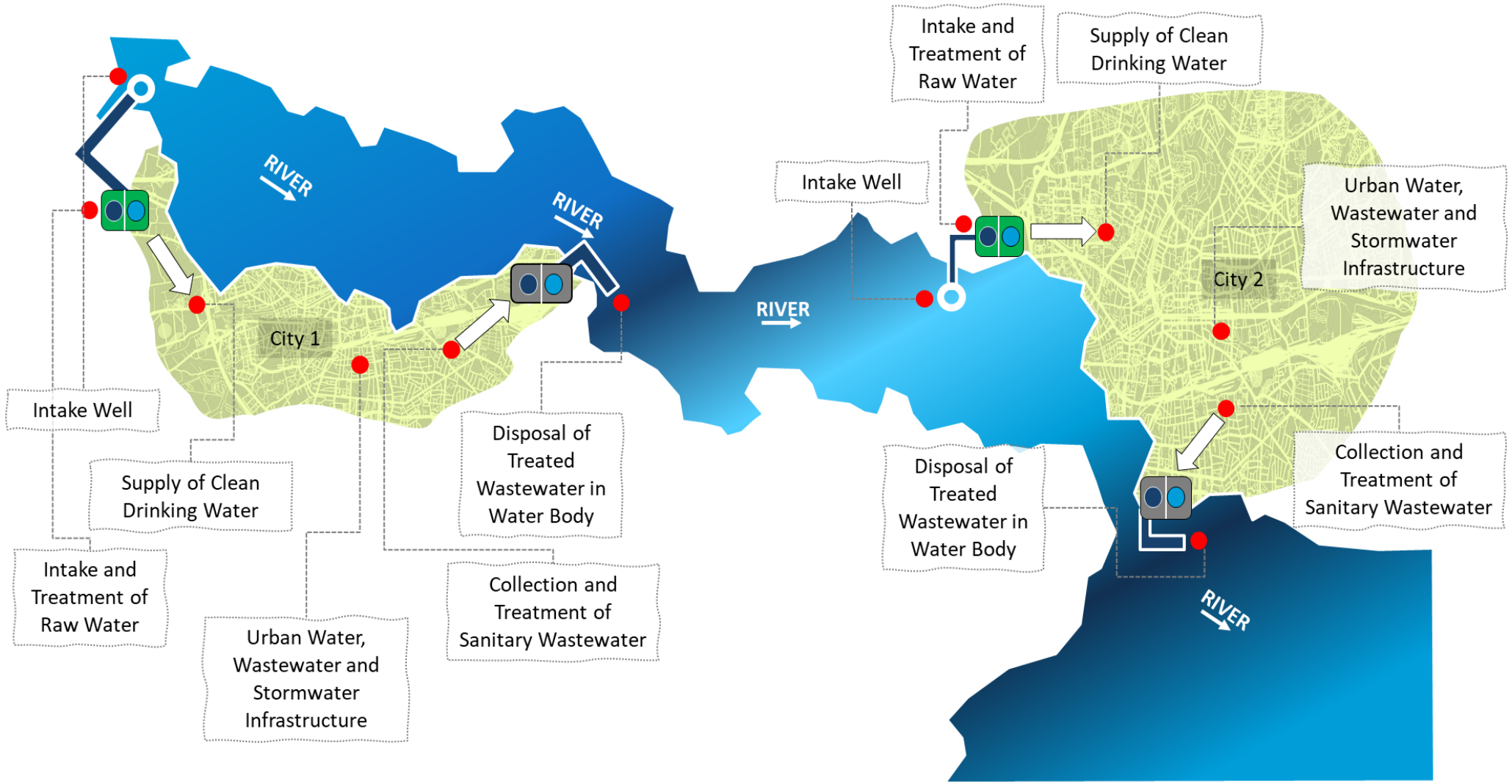


Image Credit: DTK Hydronet Solutions

Water Distribution Infrastructure

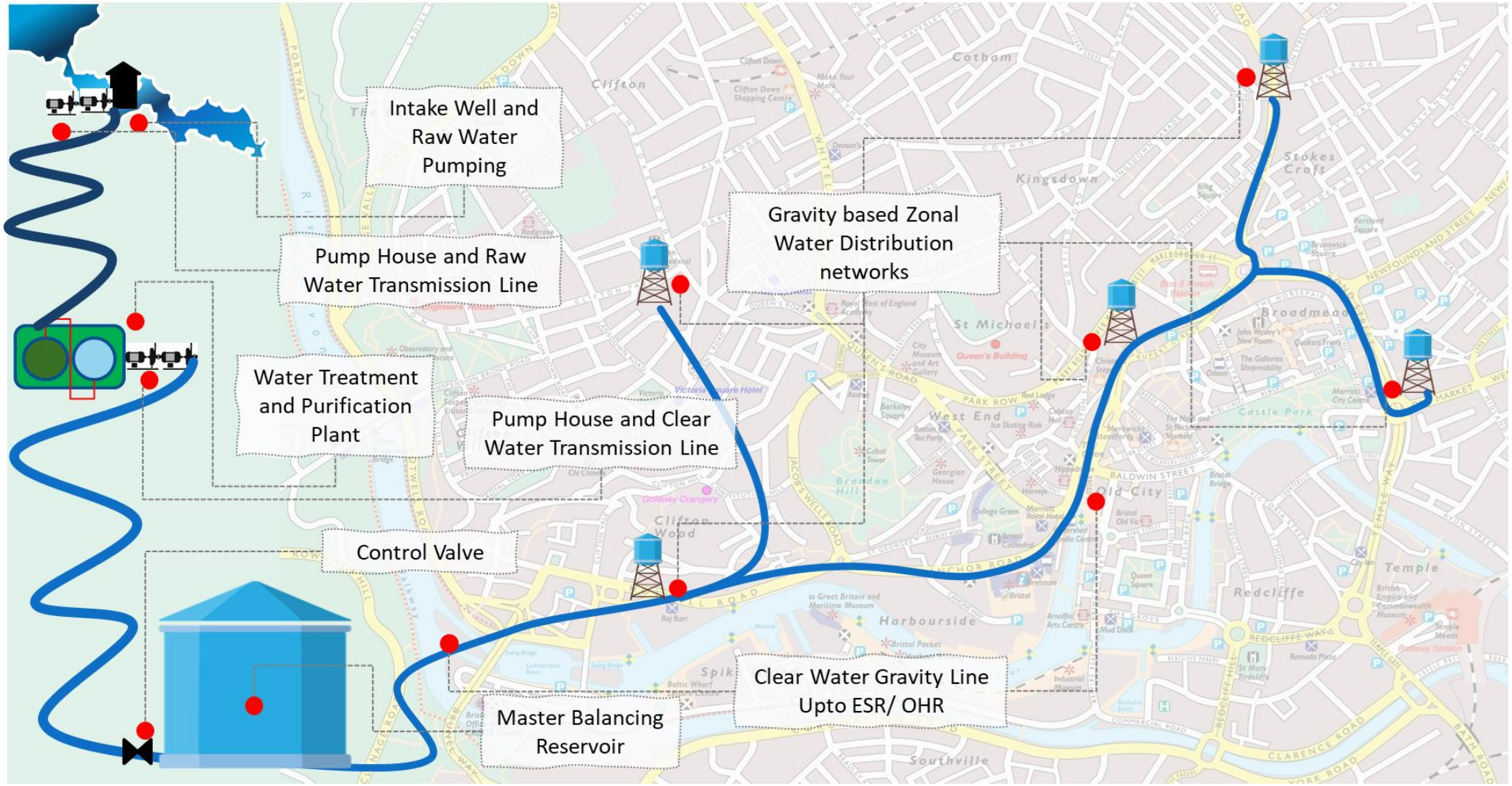


Image Credit: DTK Hydronet Solutions

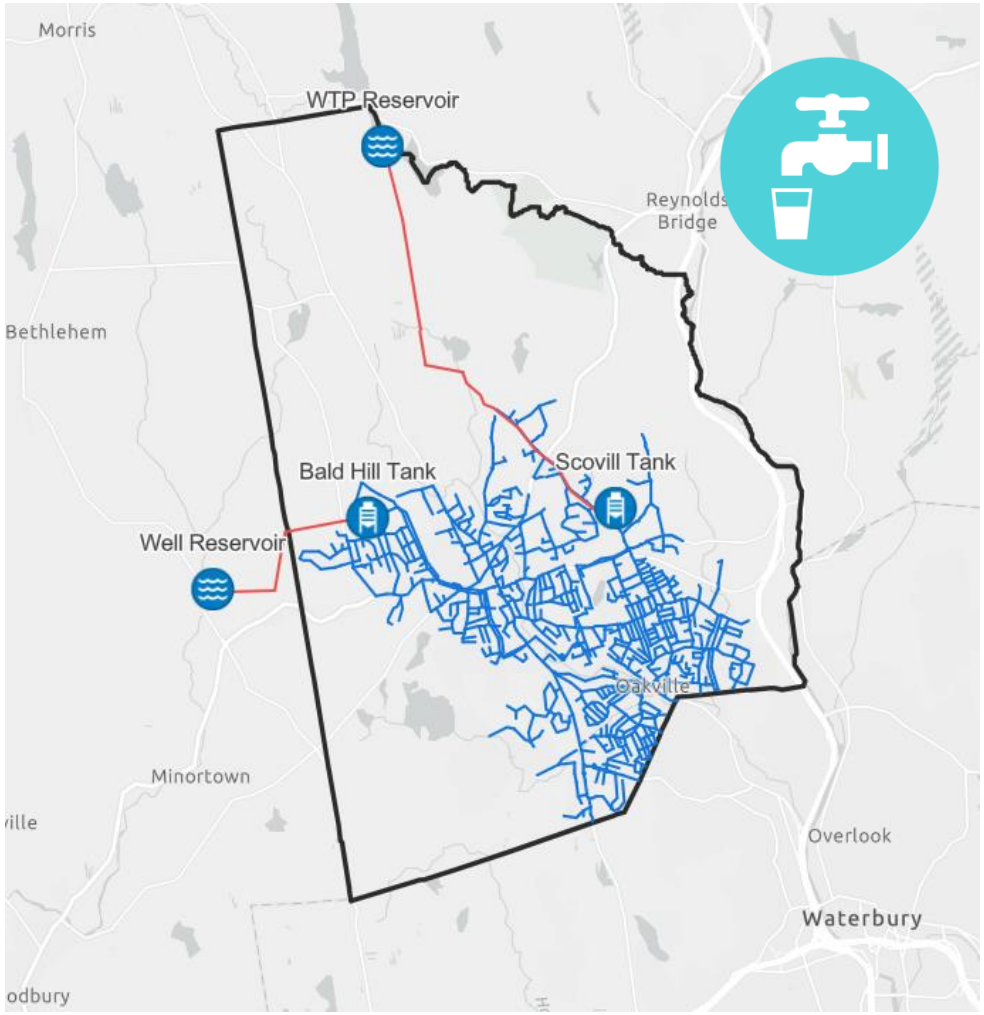
Necessity for the Project

2021
REPORT CARD
FOR AMERICA'S INFRASTRUCTURE

- America's water infrastructure is aging!
- ASCE's infrastructure report card: **C-** grade!
- USA has over 2.2 million miles of water pipes!
- 12,000 Miles replaced in 2020 alone!
- A pipe break every 2 minutes!
- 6 billion gallons of treated water lost every day!
  - That is 9000 Olympics sized swimming pools!

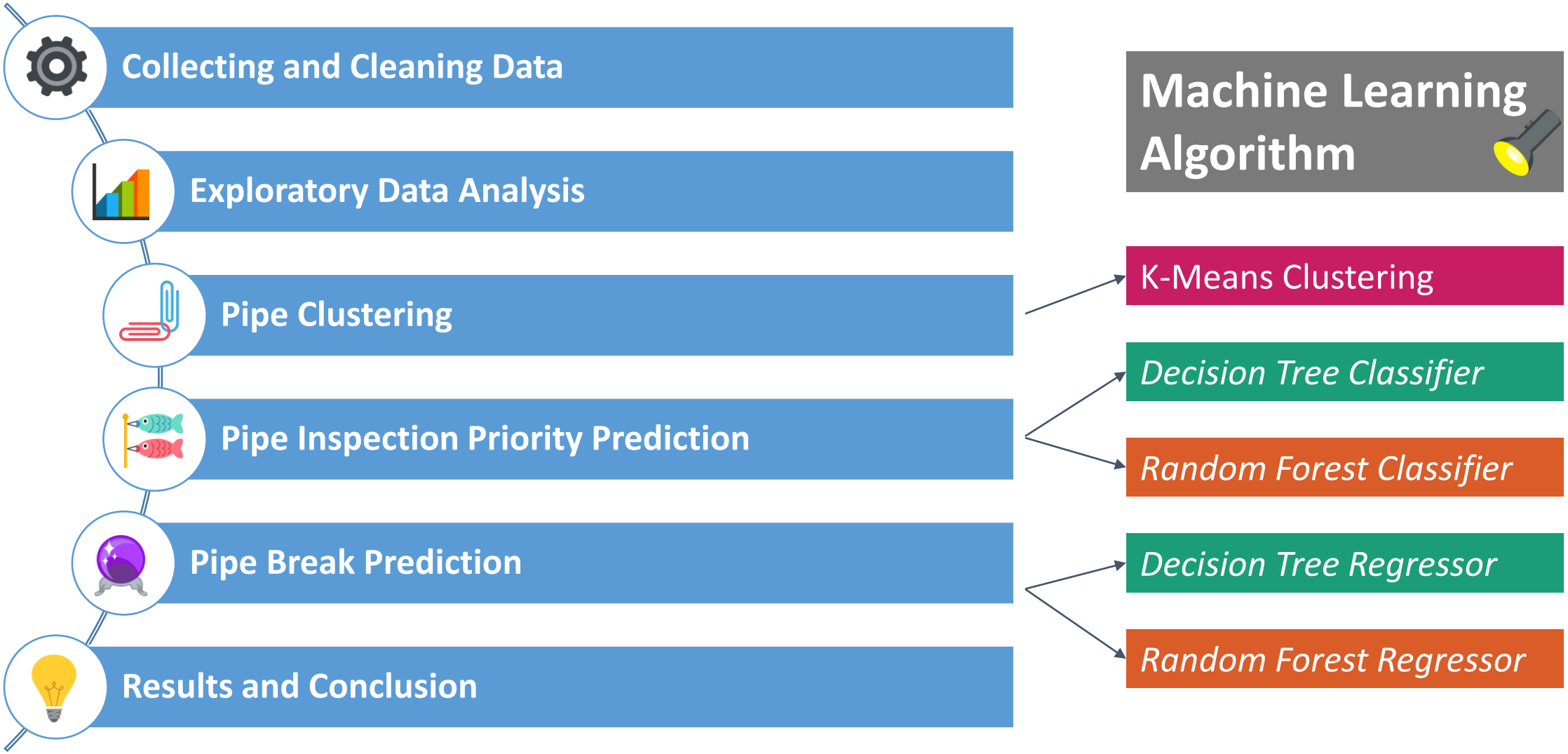
Watertown, Connecticut

- Data provided by Bentley Systems Inc.
- 110 miles of pipe network
- Pipe age – 40 to 112 years
- Network divided in 6 zones
- Caters to 6610 customers
- Population – 22,000 souls

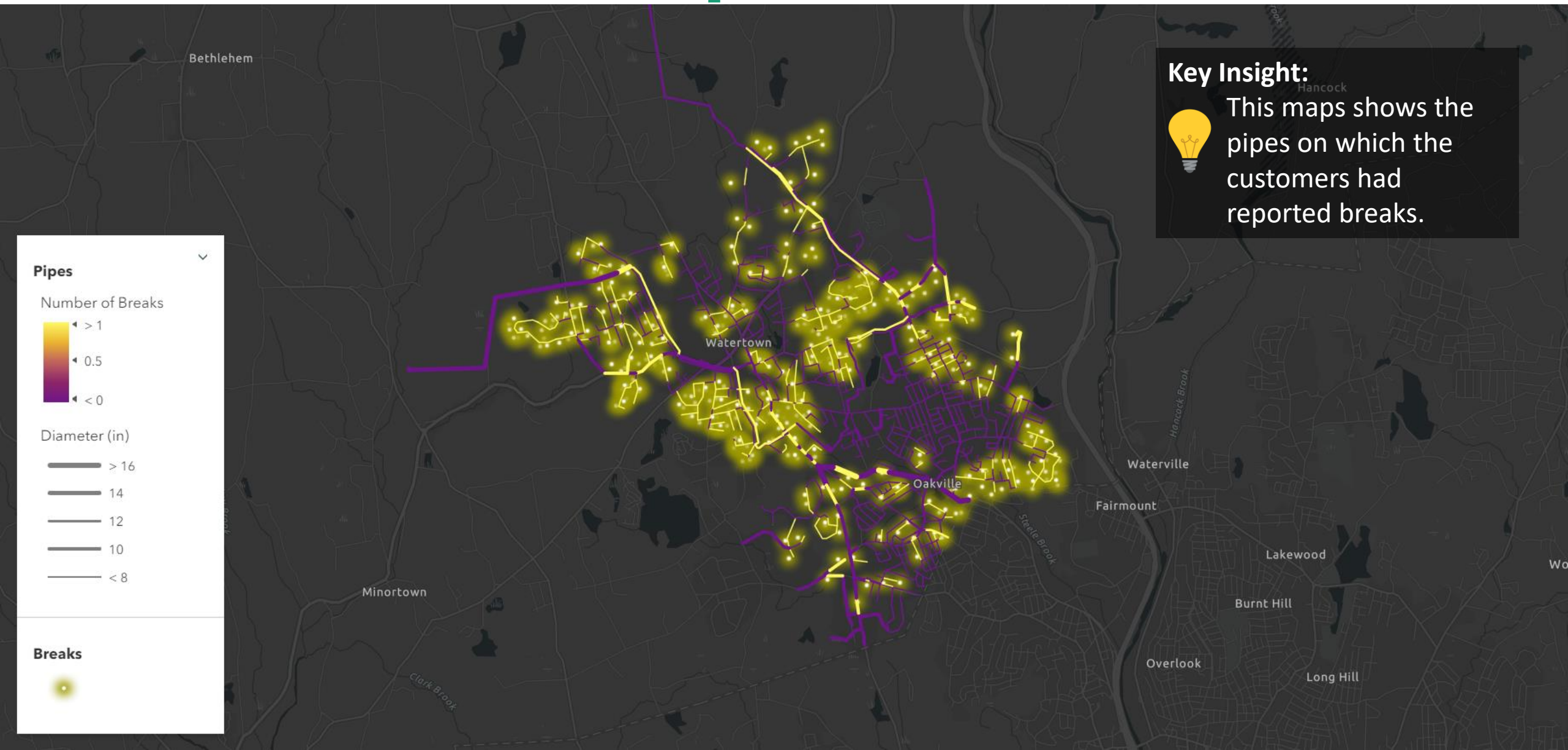


Map: Watertown Water Pipes

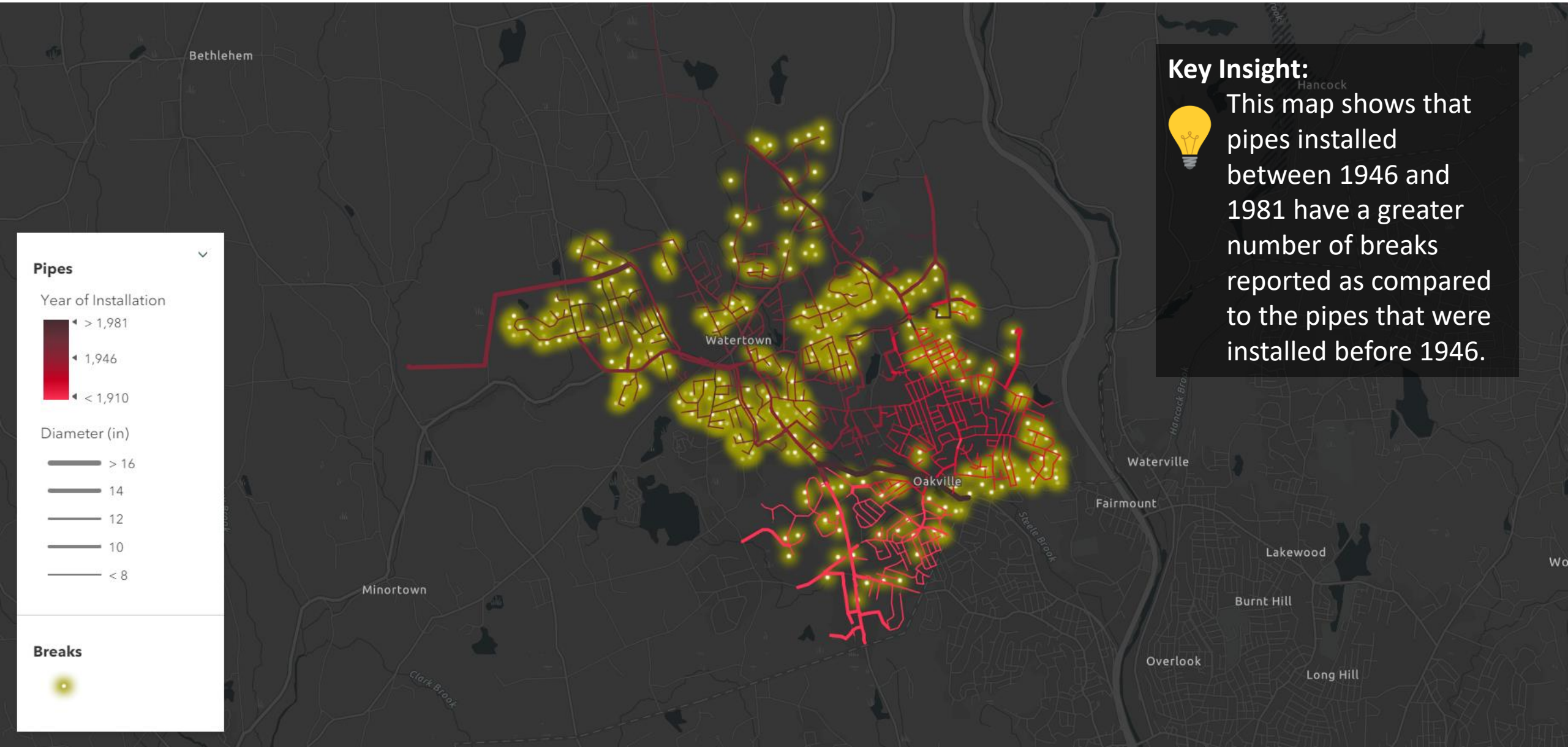
Project Methodology



The Broken Pipes!



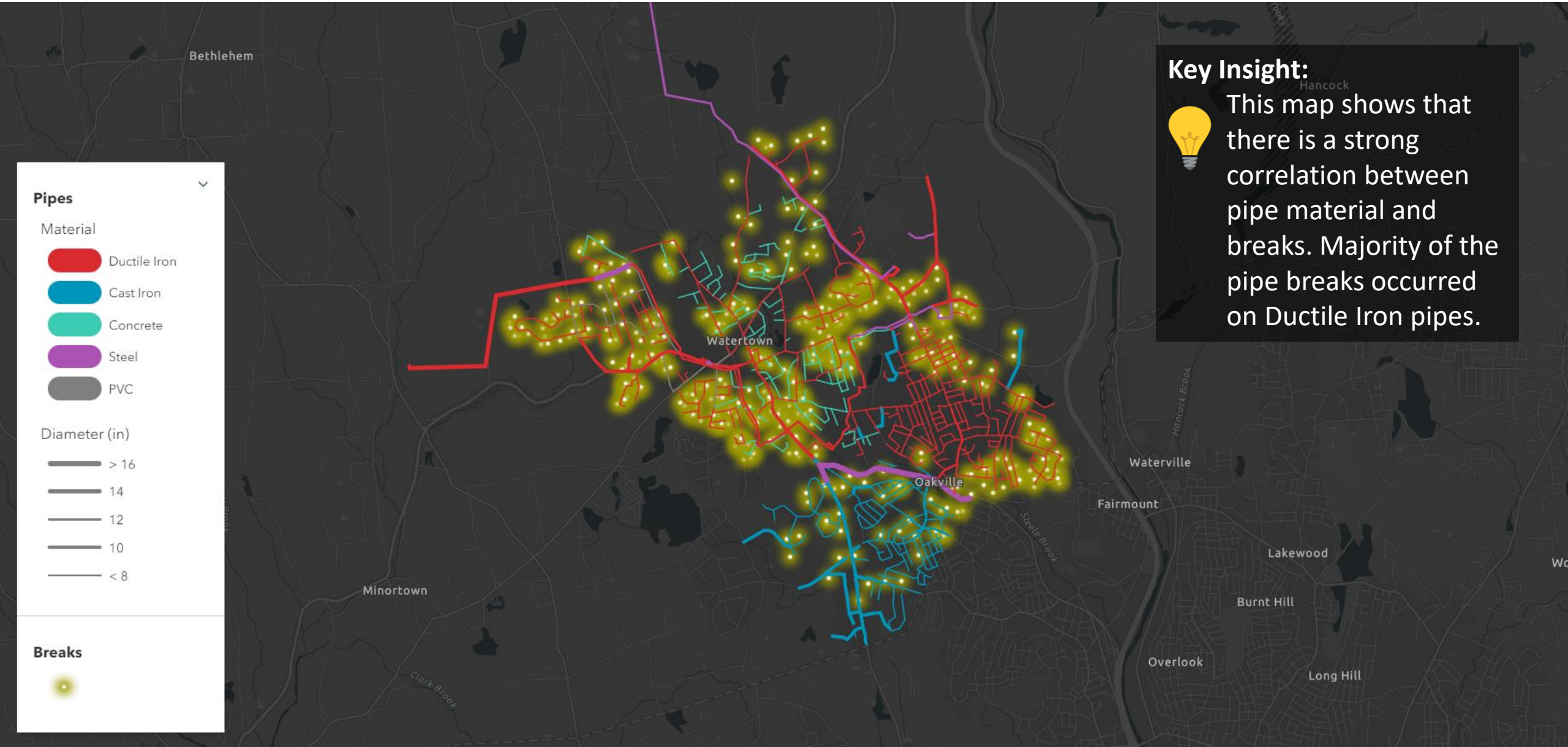
Corr: Breaks ⇆ Installation Year



Key Insight:

This map shows that pipes installed between 1946 and 1981 have a greater number of breaks reported as compared to the pipes that were installed before 1946.

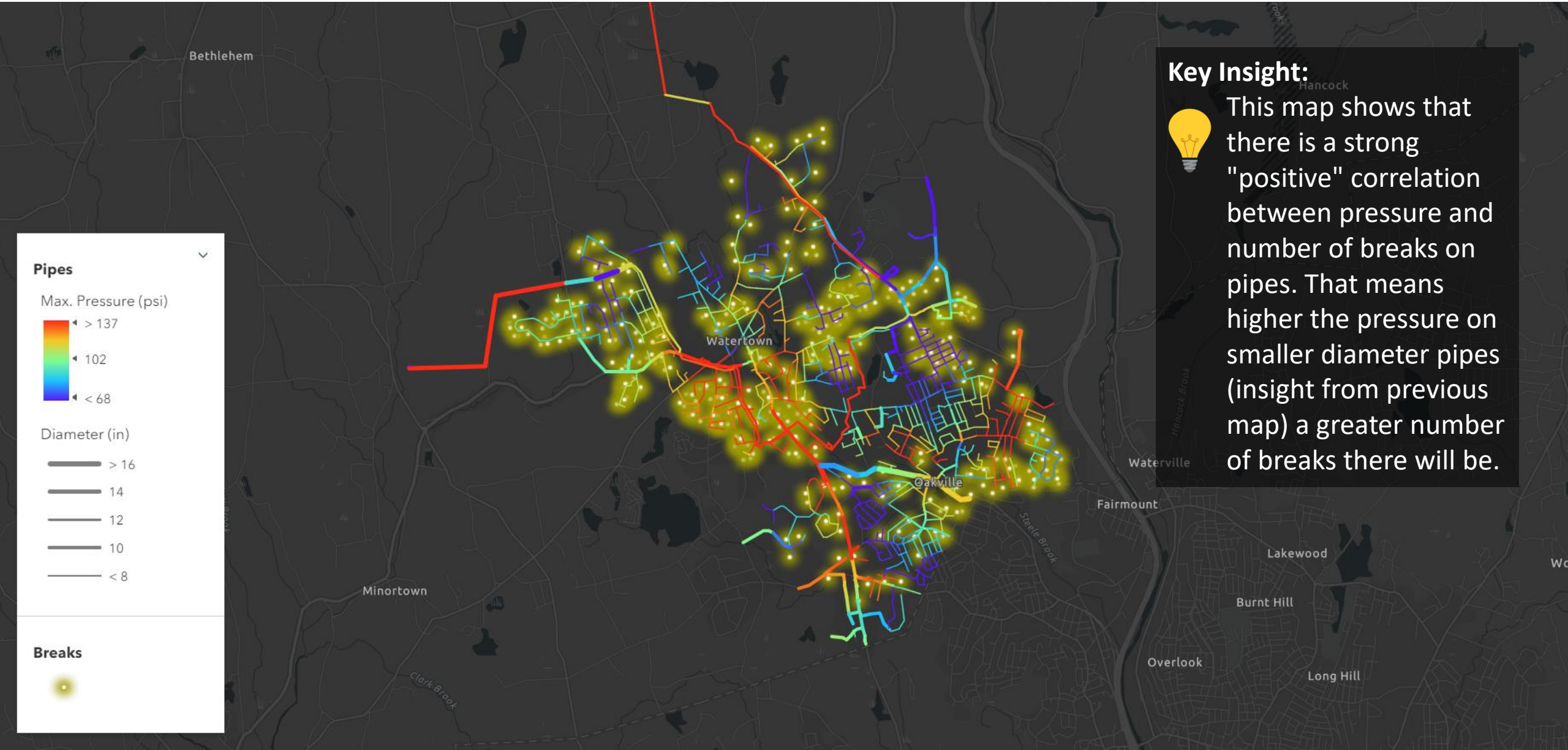
Corr: Breaks ⇆ Material



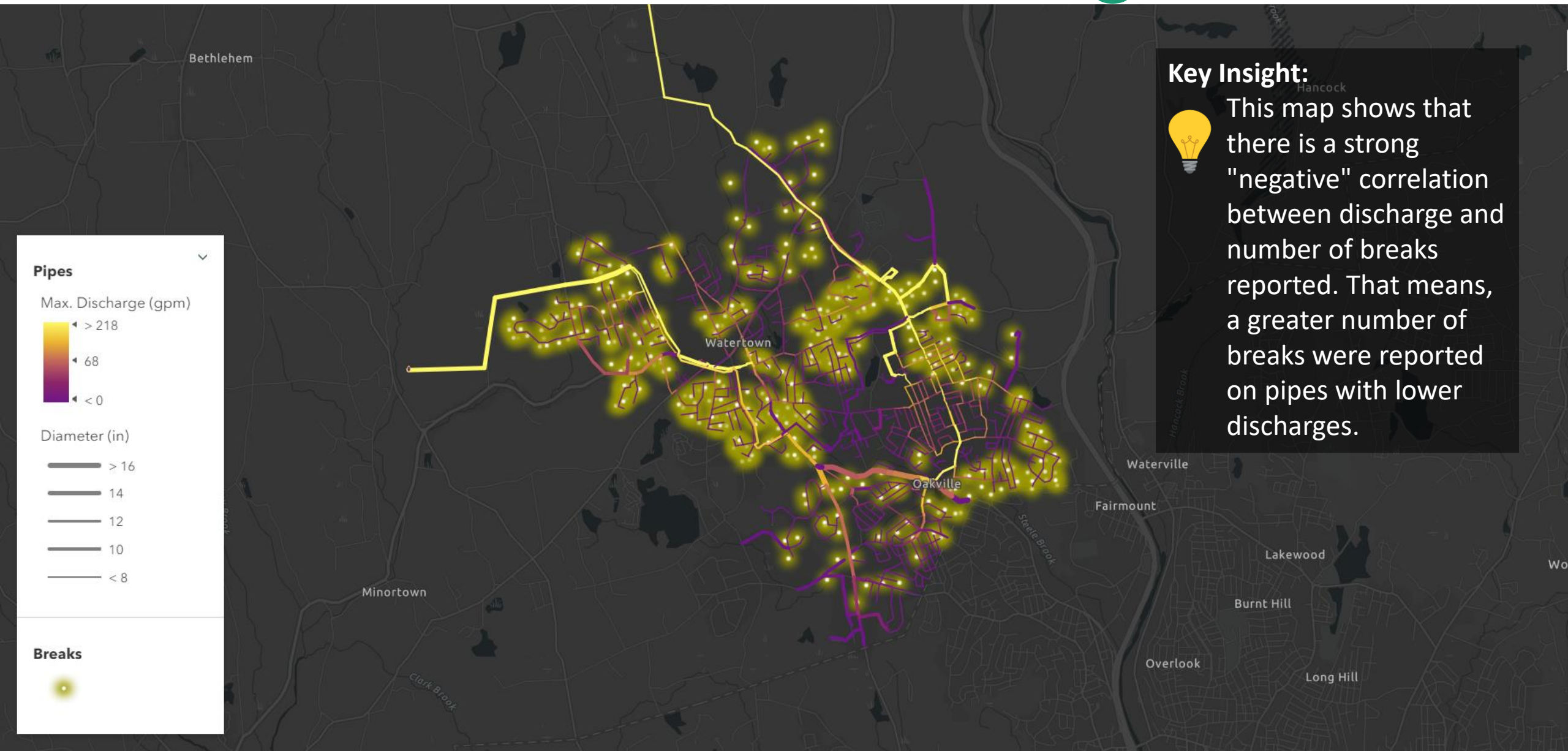
Key Insight:

This map shows that there is a strong correlation between pipe material and breaks. Majority of the pipe breaks occurred on Ductile Iron pipes.

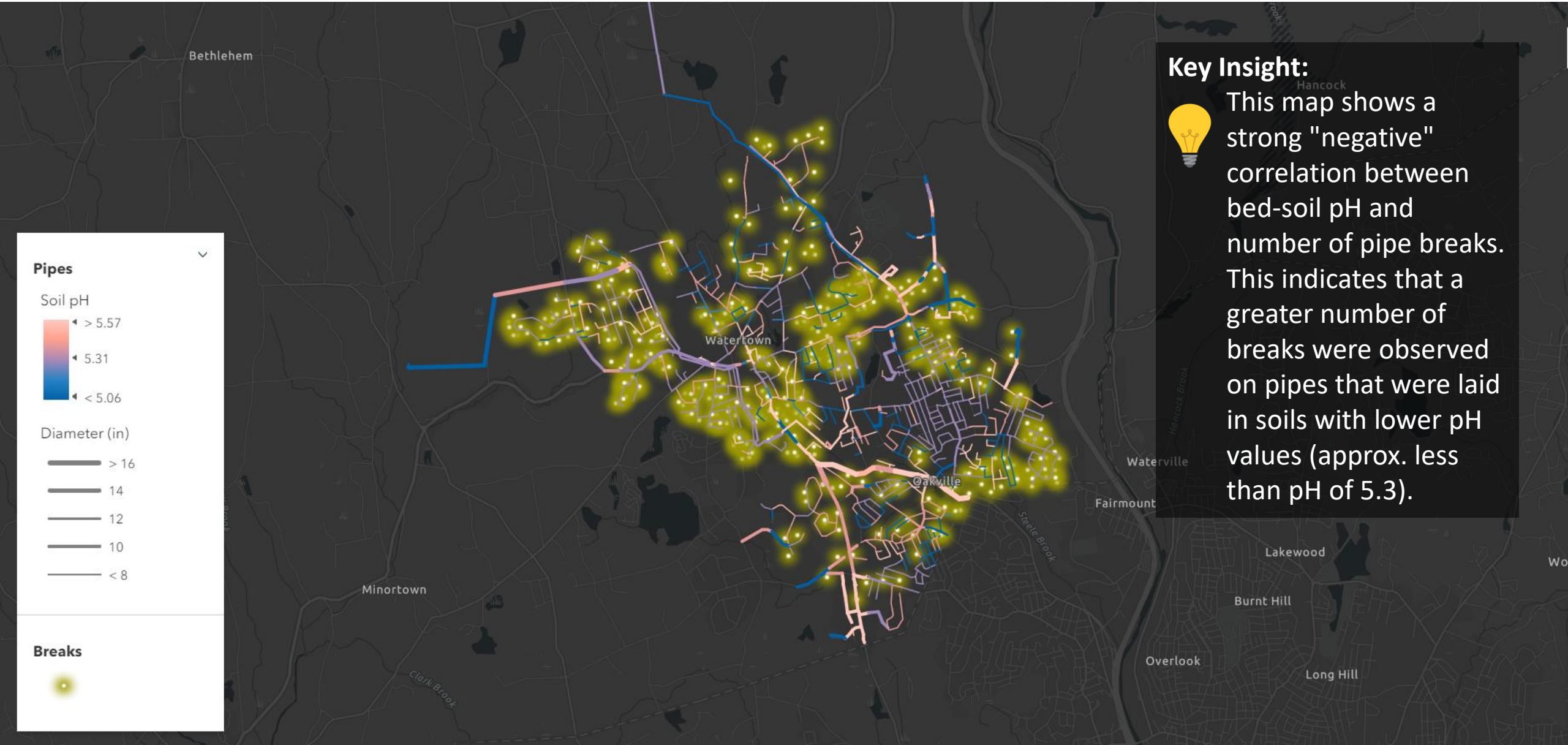
Corr: Breaks ⇆ Pressure



Corr: Breaks ⇆ Discharge



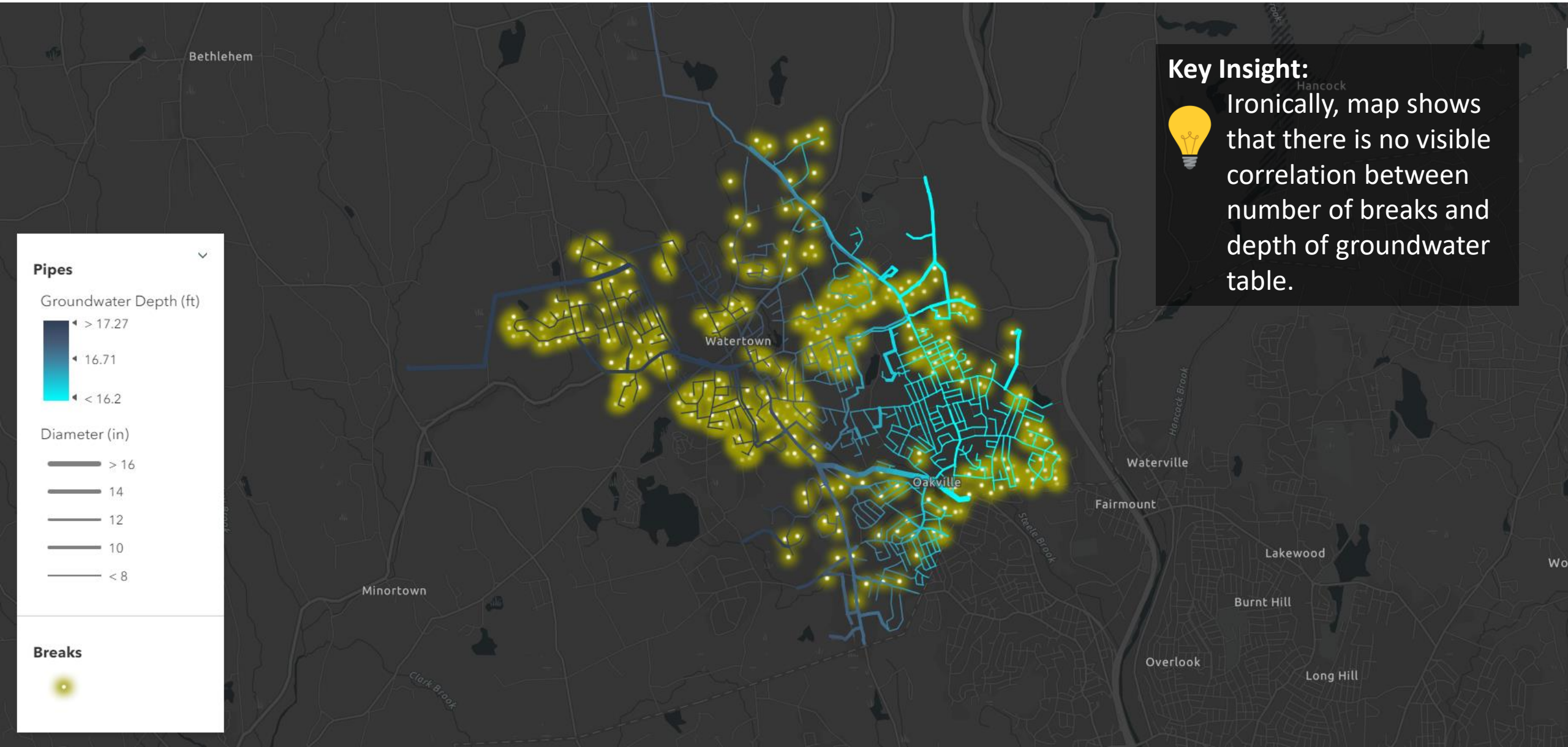
Corr: Breaks ⇆ Bed-soil pH



Key Insight:

This map shows a strong "negative" correlation between bed-soil pH and number of pipe breaks. This indicates that a greater number of breaks were observed on pipes that were laid in soils with lower pH values (approx. less than pH of 5.3).

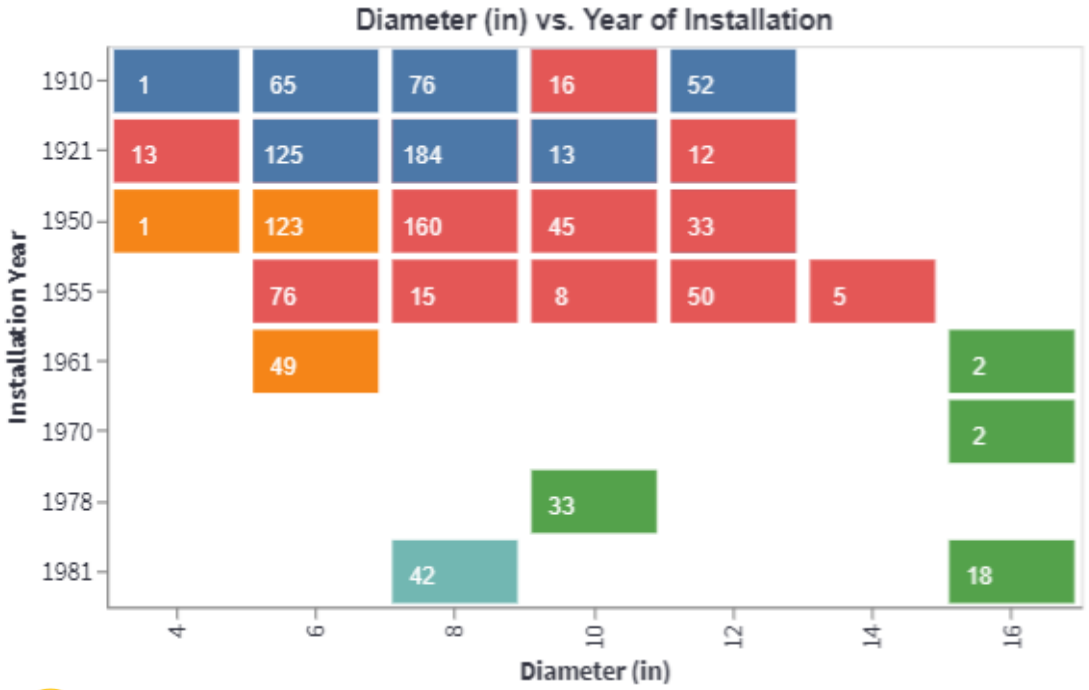
Corr: Breaks ⇆ Gr. Water Depth

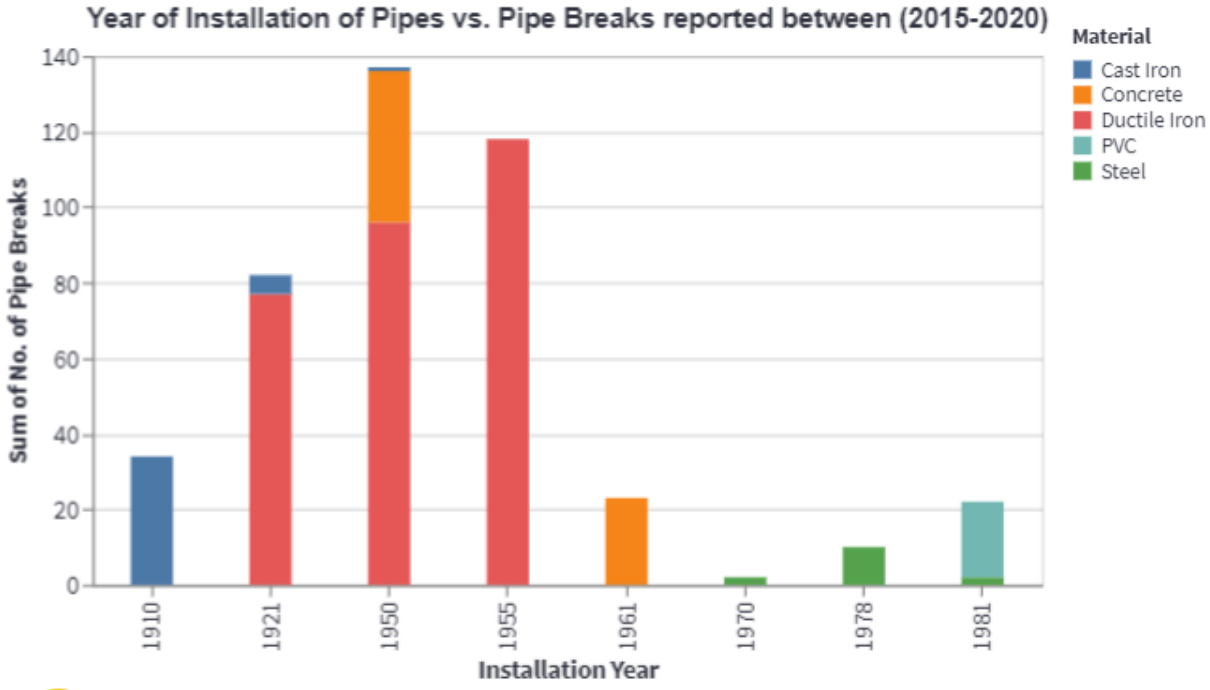


Key Insight:

Ironically, map shows that there is no visible correlation between number of breaks and depth of groundwater table.

Exploratory Data Visualization

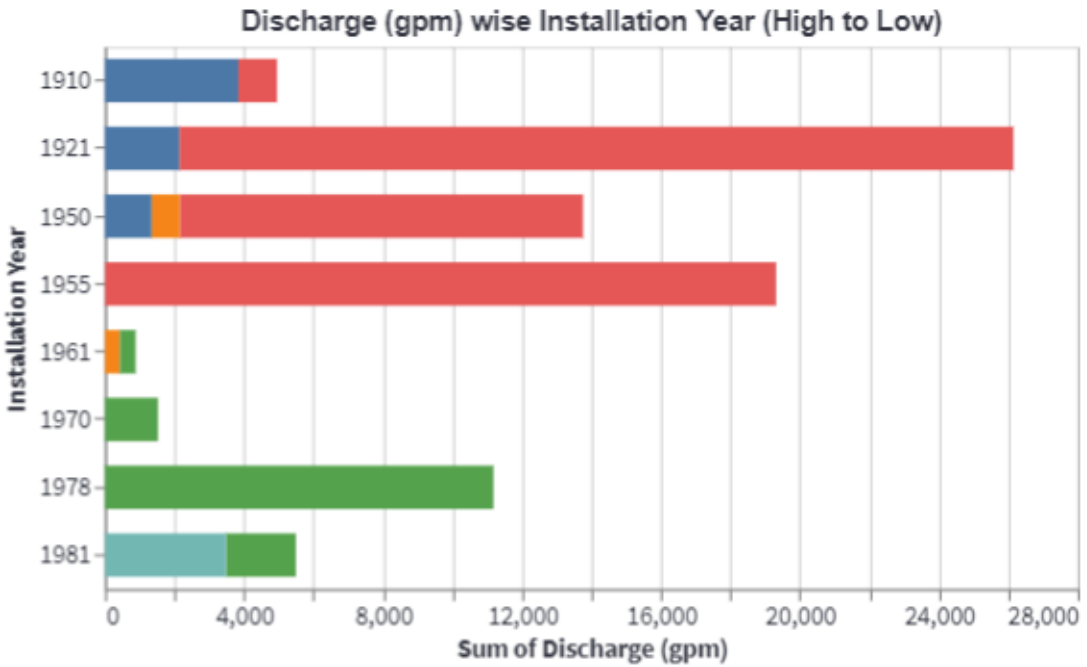




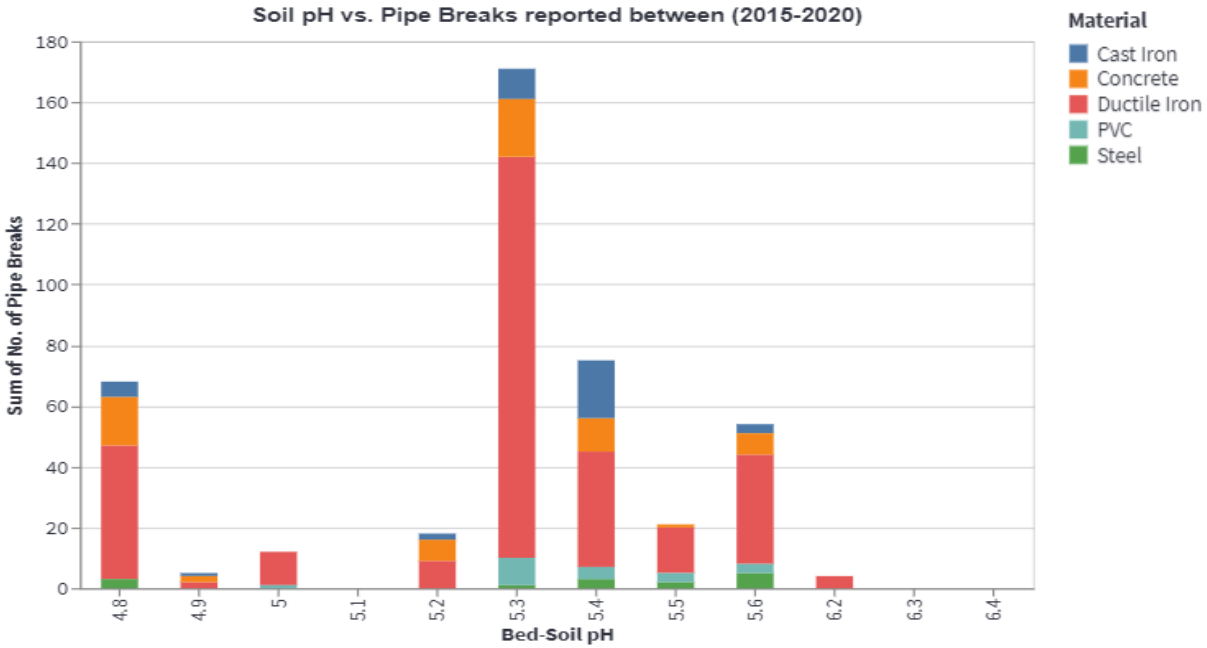
Key Insight:
Maximum number of pipes are older than 70 years and are of Ductile Iron pipes!

Key Insight:
Older ductile iron pipes have maximum records of breaks!

Exploratory Data Visualization

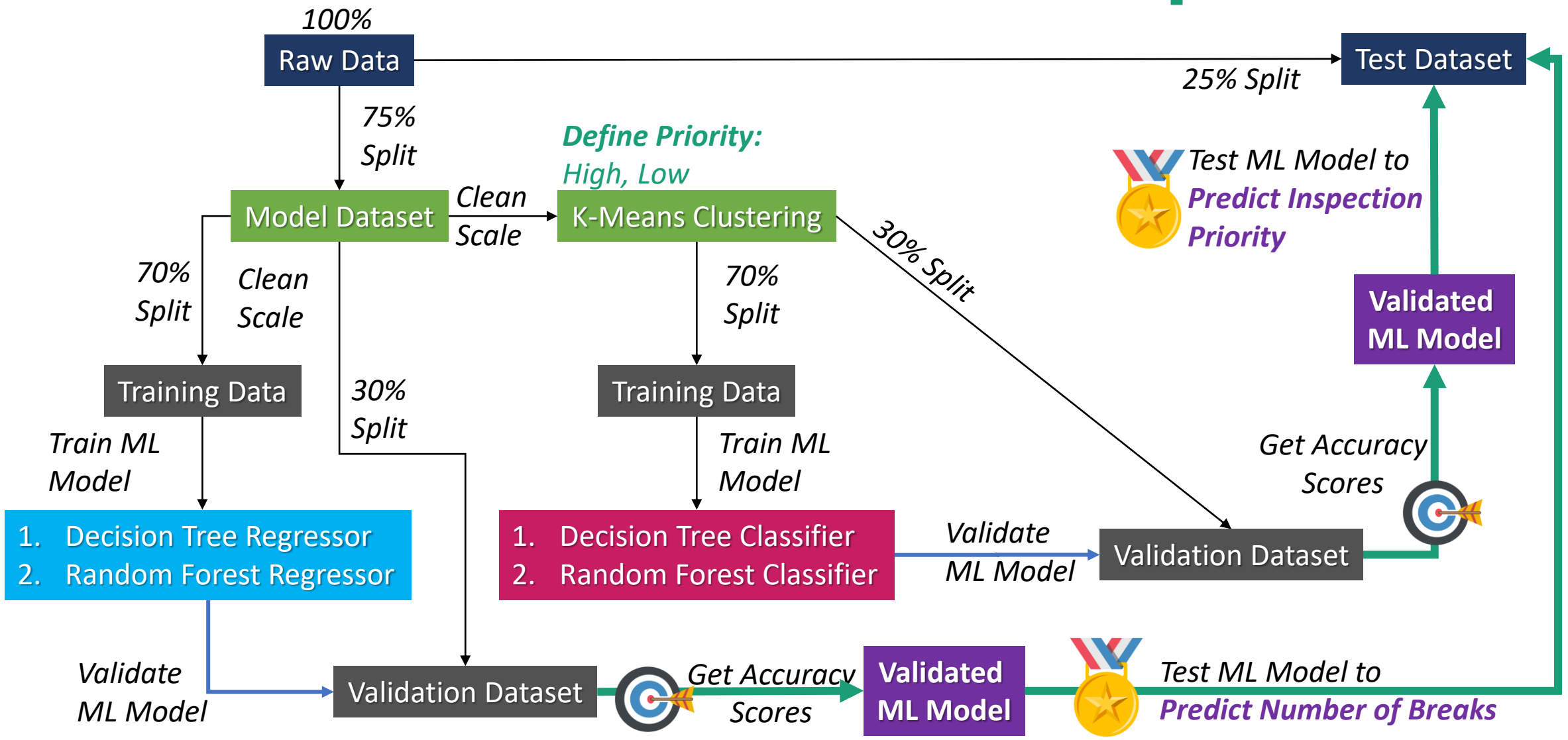


Key Insight:
Older pipes carry maximum discharge in the network! Should anything happen to these pipes, maximum number of customers would be effectively impacted!



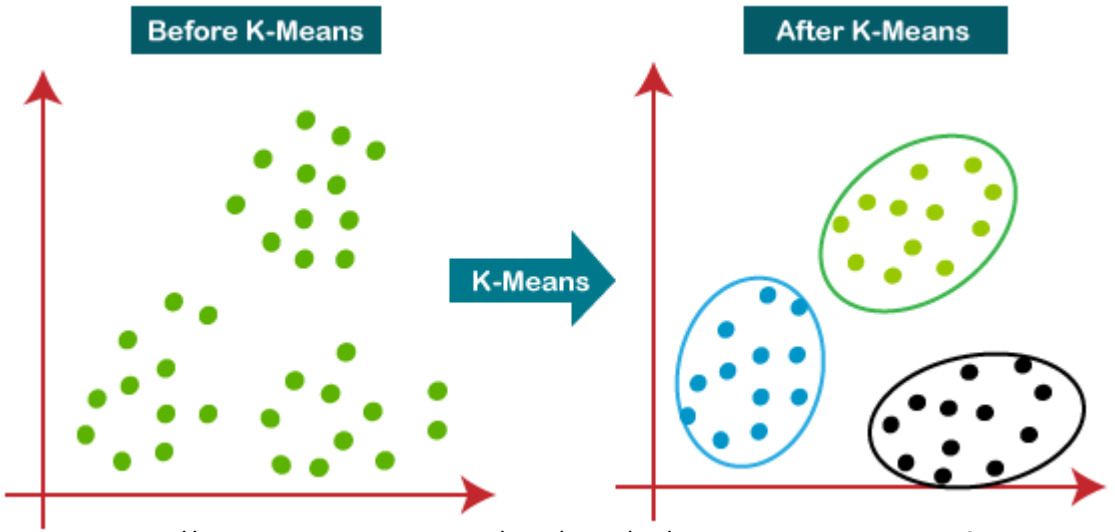
Key Insight:
pH < 6 results in higher corrosion and the number of breaks reported are more in pipes with bed-soil pH less than 6!

Data Science: Process Adopted



K-Means Clustering: Introduction

- Most popular unsupervised machine learning algorithms
- Used for solving classification problems
- K-Means segregates the unlabeled data into various groups, called clusters, based on having similar features, common patterns
- Distance based



Source: https://www.analyticsvidhya.com/blog/2021/04/k-means-clustering-simplified-in-python/

K-Means Clustering: Pre-process

Dataframe

Diameter	Length	Material	Groundwater Depth	Installation Year	Age	Pressure	Discharge	Soil pH	Breaks No.

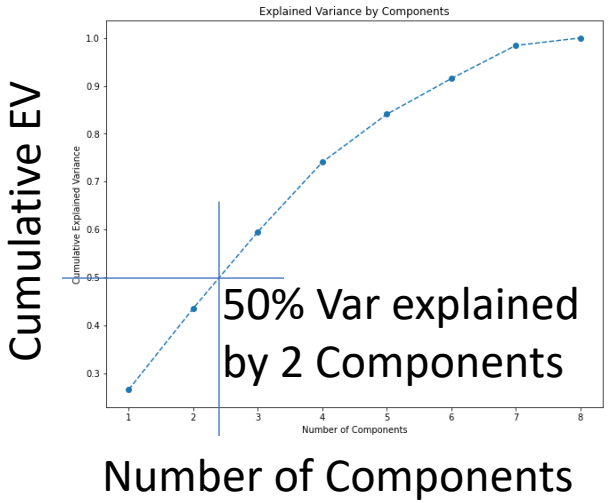
Label Encoding (0, 1, 2, …)

Material	Encoded
Ductile Iron	0
Cast Iron	1
PVC	2
Ductile Iron	0
Steel	3

Standardization (0 to1)

Age	Diameter
0.001	0.02
0.002	0.01
0.014	0.02
0.012	0.06
0.015	0.04

Explained Variance (EV)

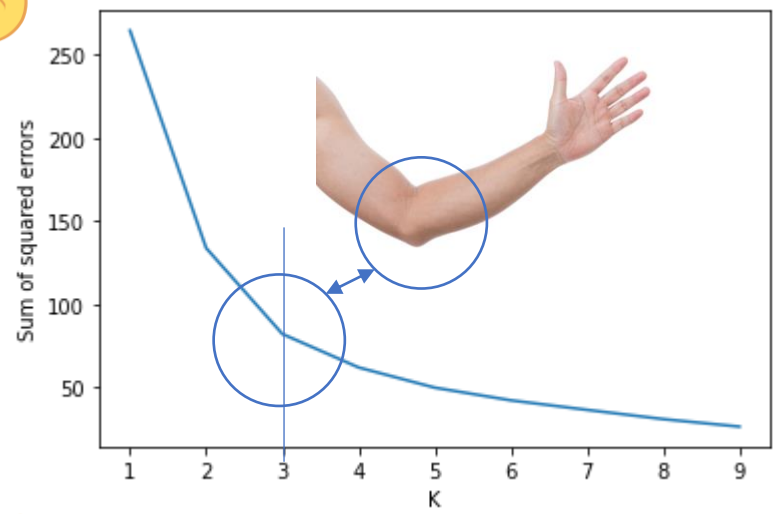


Dimensionality Reduction

PCA 1	PCA 2
0.31	0.22
0.02	0.41
0.014	0.02
0.012	0.5
0.015	0.45

K-Means Clustering: # of Clusters

Elbow Method

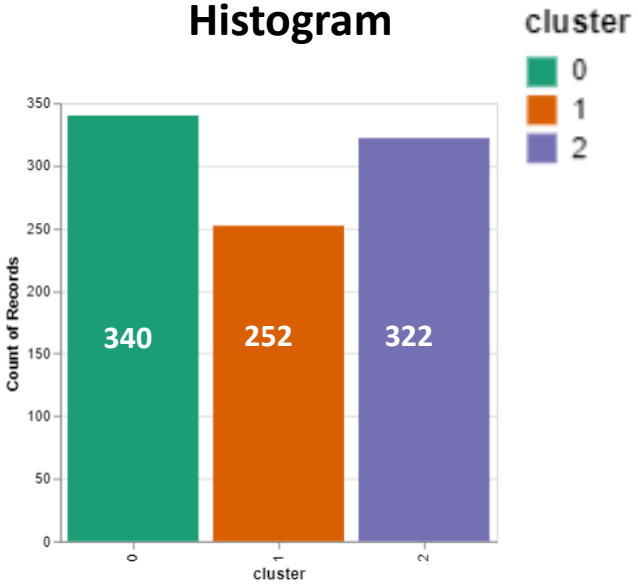


# of Clusters = 3

Dataframe with Cluster#

ID	Diameter	LENGTH_FT	Breaks_No	Age	Ncustomers	PH	Pmax_Psi	Qmax_gpm	cluster
36001	0.50	0.245573	0.0	0.84507	0.229167	0.500000	0.571429	1.000000	1
36460	0.50	1.000000	0.0	1.00000	0.875000	0.666667	0.595238	0.074967	0
30896	0.25	0.874346	0.0	0.84507	0.604167	1.000000	0.833333	0.021419	0
34424	0.50	0.700287	0.0	0.43662	0.020833	1.000000	0.412698	0.000000	2
34052	0.50	0.543768	0.0	0.84507	0.125000	0.000000	0.817460	0.096386	0

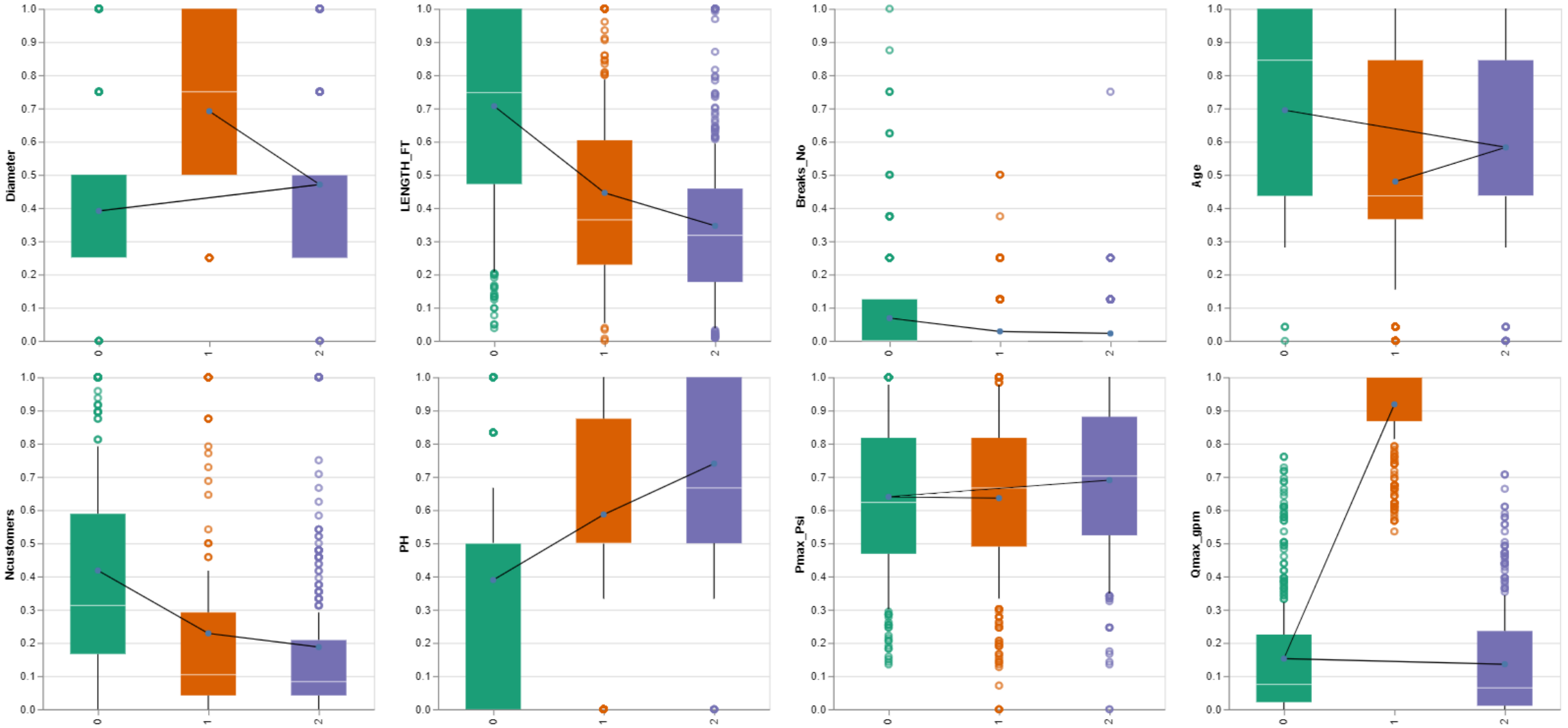
Histogram



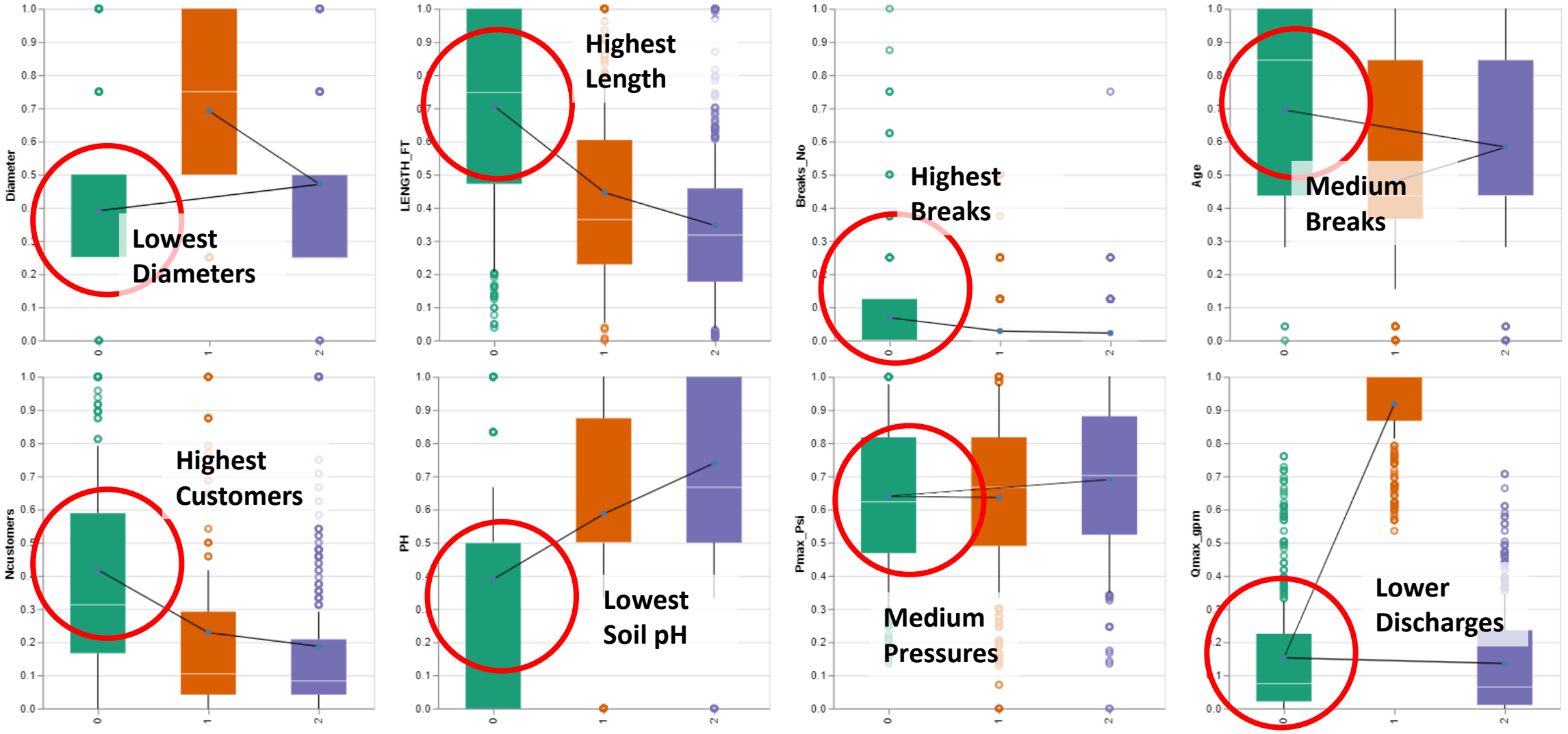
Silhouette Score Method

Silhouette Score for k(clusters) = 2 is 0.49
Silhouette Score for k(clusters) = 3 is 0.44
Silhouette Score for k(clusters) = 4 is 0.42
Silhouette Score for k(clusters) = 5 is 0.36
Silhouette Score for k(clusters) = 6 is 0.38
Silhouette Score for k(clusters) = 7 is 0.37
Silhouette Score for k(clusters) = 8 is 0.37
Silhouette Score for k(clusters) = 9 is 0.38

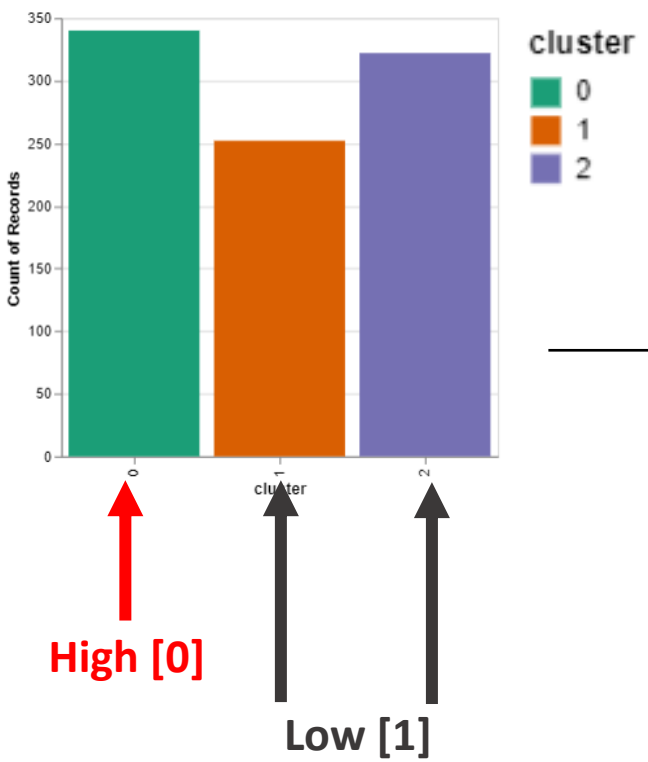
Understanding the Clusters



Understanding the Clusters



Defining “Inspection Priority”



Label	Diameter	Area	ENGTH_F	3reaks_Nc	Age	Icustomer	PH	Pmax_Psi	)max_gpn	cluster	Condition
P-1022	8	50.24	190	0	101	11	5.3	89	172	1	1
P-488	8	50.24	2108	0	112	42	5.4	92	7	0	0
P-407	6	28.26	656	0	101	29	5.6	122	2	0	0
P-108	8	50.24	527	0	72	1	5.6	69	0	2	1
P-589	8	50.24	411	0	101	6	4.9	120	9	0	0
P-197	12	113.04	222	0	112	9	5.6	101	2	2	1
P-440	6	28.26	668	8	67	15	5.3	135	3	0	0
P-443	6	28.26	245	0	61	4	5.4	87	9	2	1

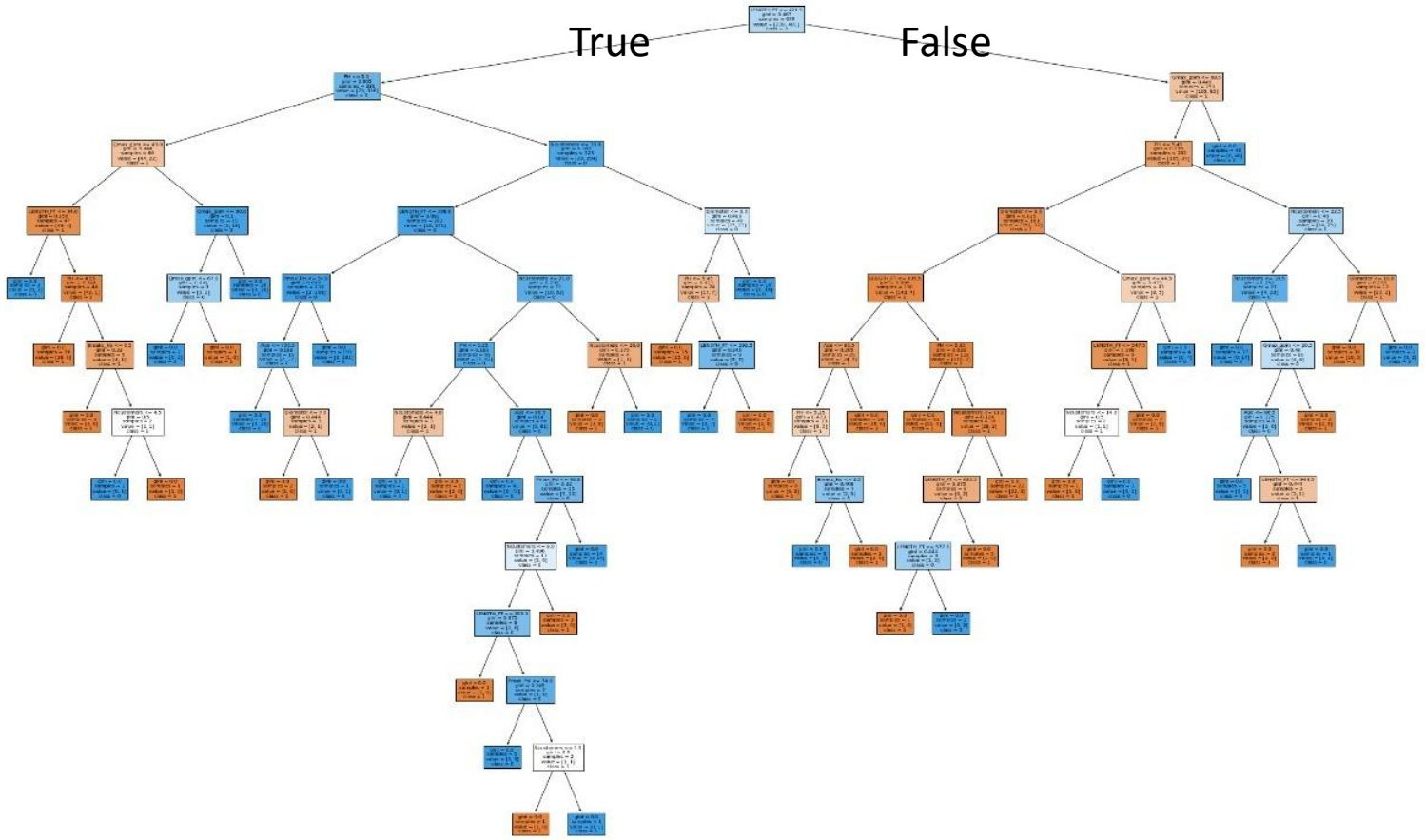
Inspection Priority = {High, Low}

= {0, 1}

Two Objectives

1. ML Classification for predicting “Inspection Priority”
2. ML Regression for predicting “Number of Breaks”

Training Decision Tree Classifier

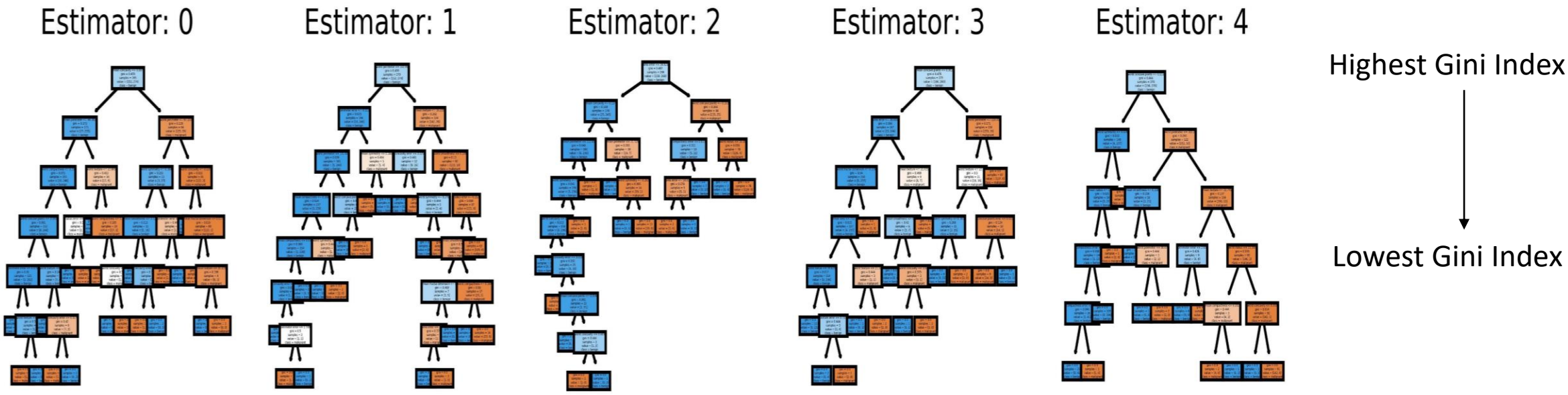


Highest Gini* Index

Lowest Gini* Index

*Gini Index is a measure of impurity of the sample. The decision tree finds the feature with highest Gini Index to begin the classification. The tree always end at Gini Index = 0. i.e., it is pure sample i.e., either it is a High Priority Pipe or a Low Priority Pipe.

Training Random Forest Classifier

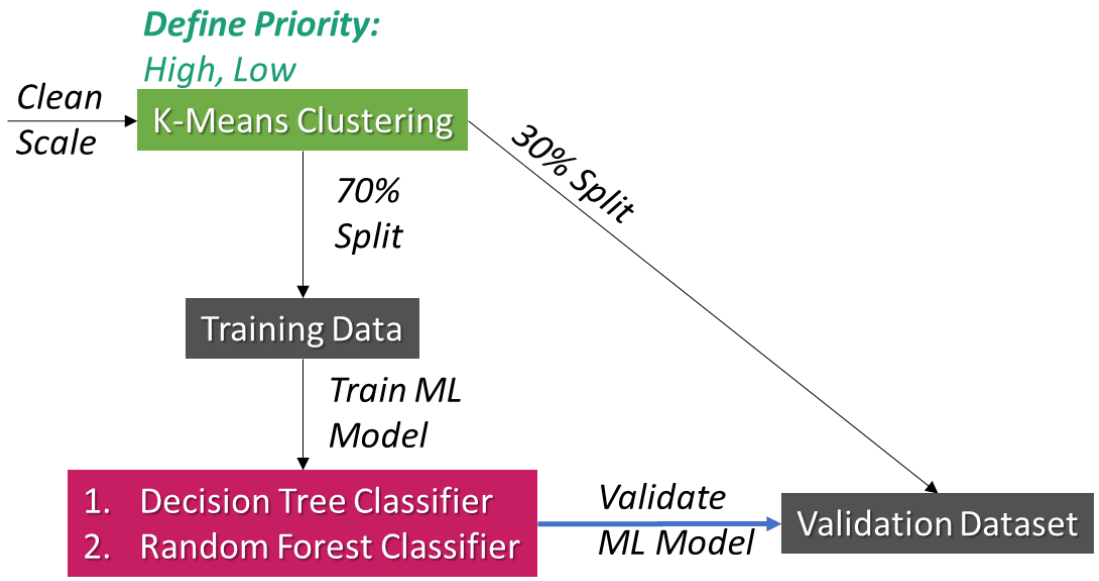


Source: https://stackoverflow.com/questions/40155128/plot-trees-for-a-random-forest-in-python-with-scikit-learn

As the name suggests, Random Forest, randomly generates numerous decision trees simultaneously and learns a pattern to predict the class. Random Forest Classifier is the most widely used ML Algorithm so far.

Training ML Classifiers

- Decision Tree Classifier
- Random Forest Classifier



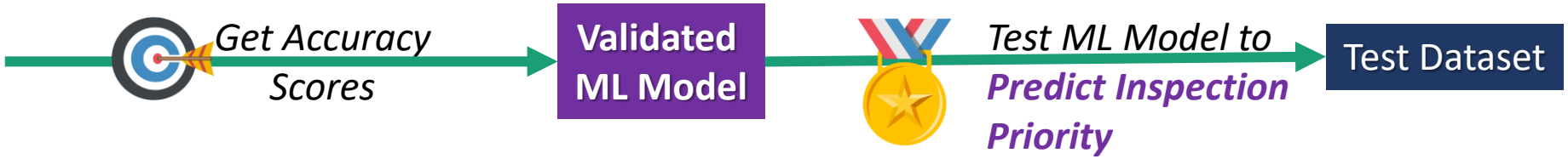

```
#1. Decision Tree Classification Model Testing
# Create Decision Tree classifer object
A = df25.Score
B = df25[features]
#dt_model1 = tree.DecisionTreeClassifier(random_state=111)
#dt_model.fit(B, A)
Pred_A_DT = DT.predict(B)

#2. Random Forest Classification  Testing
# Create Decision Tree classifer object
Pred_A_RF = RF.predict(B)

#Exporting combined results
df=pd.DataFrame()
df = df25
df['Score DT'] = Pred_A_DT
df['Score RF'] = Pred_A_RF
#df.rename(columns={'0': 'Score'}, inplace=True)
df['Inspection Priority by DT'] = ""
df['Inspection Priority by RF'] = ""
df['Inspection Priority by DT'] = np.where(df['Score DT']== '0', 'High', 'Low')
df['Inspection Priority by RF'] = np.where(df['Score RF']== '0', 'High', 'Low')
```

Validation on *Validation Dataset*

	DECISION TREE CLASSIFICATION	RANDOM FOREST CLASSIFICATION
MEAN SQUARED ERROR	0.069	0.112
ACCURACY	93.29%	88.72%
CONFUSION MATRIX	[87 15] [4 169]	[84 18] [13 160]



Result: Test Dataset, Insp. Priority

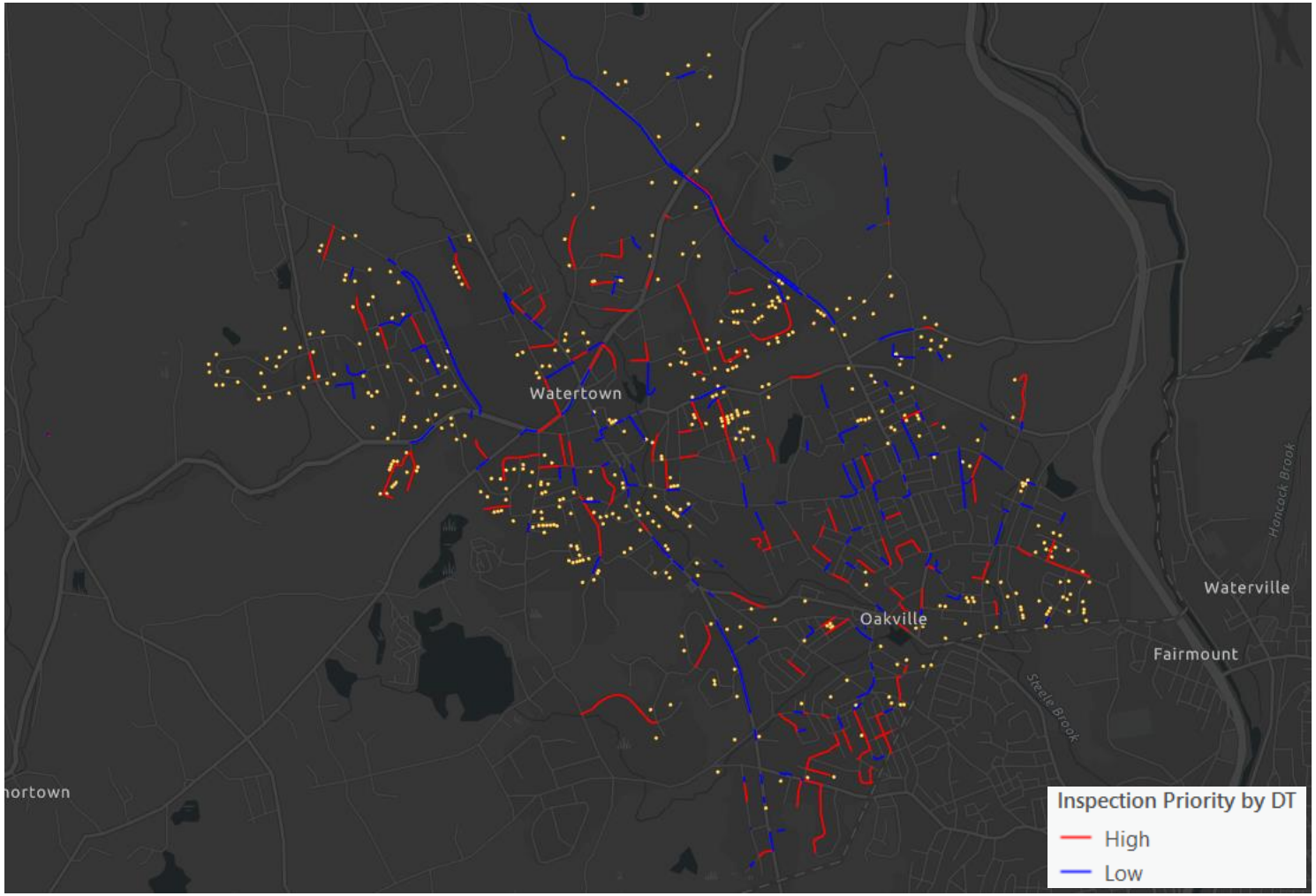
- ML Classifiers applied to Test Dataset
- Inspection Priorities predicted

ID	Label	Diameter	LENGTH_FT	Breaks_No	Age	Ncustomers	PH	Pmax_Psi	Qmax_gpm	Inspection Priority by DT	Inspection Priority by RF
29605	P-541	4	276	0	101	53	4.8	133	2	High	High
29964	P-502	4	343	0	101	11	4.8	102	2	High	High
31123	P-308	8	1028	7	72	16	4.8	34	2	High	High
33440	P-362	6	135	0	72	9	4.8	127	2	High	High
33443	P-25	8	26	0	101	0	4.8	35	0	Low	Low
33476	P-3	6	456	0	101	24	4.8	98	0	High	High
33577	P-652	6	252	0	61	15	4.8	65	21	High	High
33582	P-610	6	323	0	72	17	4.8	103	6	High	High
33803	P-369	4	188	0	101	13	4.8	95	1	High	High
33896	P-455	6	803	3	72	13	4.8	108	4	High	High
33917	P-304	12	440	0	67	19	4.8	67	5	Low	High
34086	P-151	6	1095	6	72	18	4.8	63	0	High	High
34134	P-625	6	651	0	101	30	4.8	102	22	High	High
34158	P-685	6	538	0	72	10	4.8	105	12	High	High
34443	P-520	6	209	0	72	7	4.8	112	5	High	High
34579	P-581	8	471	0	101	53	4.8	122	11	High	High
34594	P-759	6	236	0	112	3	4.8	68	15	High	High
34918	P-514	8	1449	2	72	34	4.8	106	8	High	High
35090	P-430	8	144	0	72	37	4.8	102	5	High	High
35126	P-667	8	766	0	72	20	4.8	104	23	High	High
35138	P-764	8	465	0	72	13	4.8	88	26	High	High
35248	P-333	12	44	0	67	22	4.8	72	6	High	High
35327	P-857	12	840	0	101	7	4.8	107	160	Low	Low
35336	P-626	12	920	1	72	136	4.8	125	33	High	High
35449	P-1096	8	450	1	72	14	4.8	48	49	High	High
35499	P-1046	10	141	0	44	7	4.8	112	220	Low	Low
35844	P-562	8	232	0	101	27	4.8	140	10	High	High
35845	P-513	8	220	0	101	22	4.8	146	8	High	High

Random Samples from Test Dataset

Insp. Priority, DT on 25% Test Data

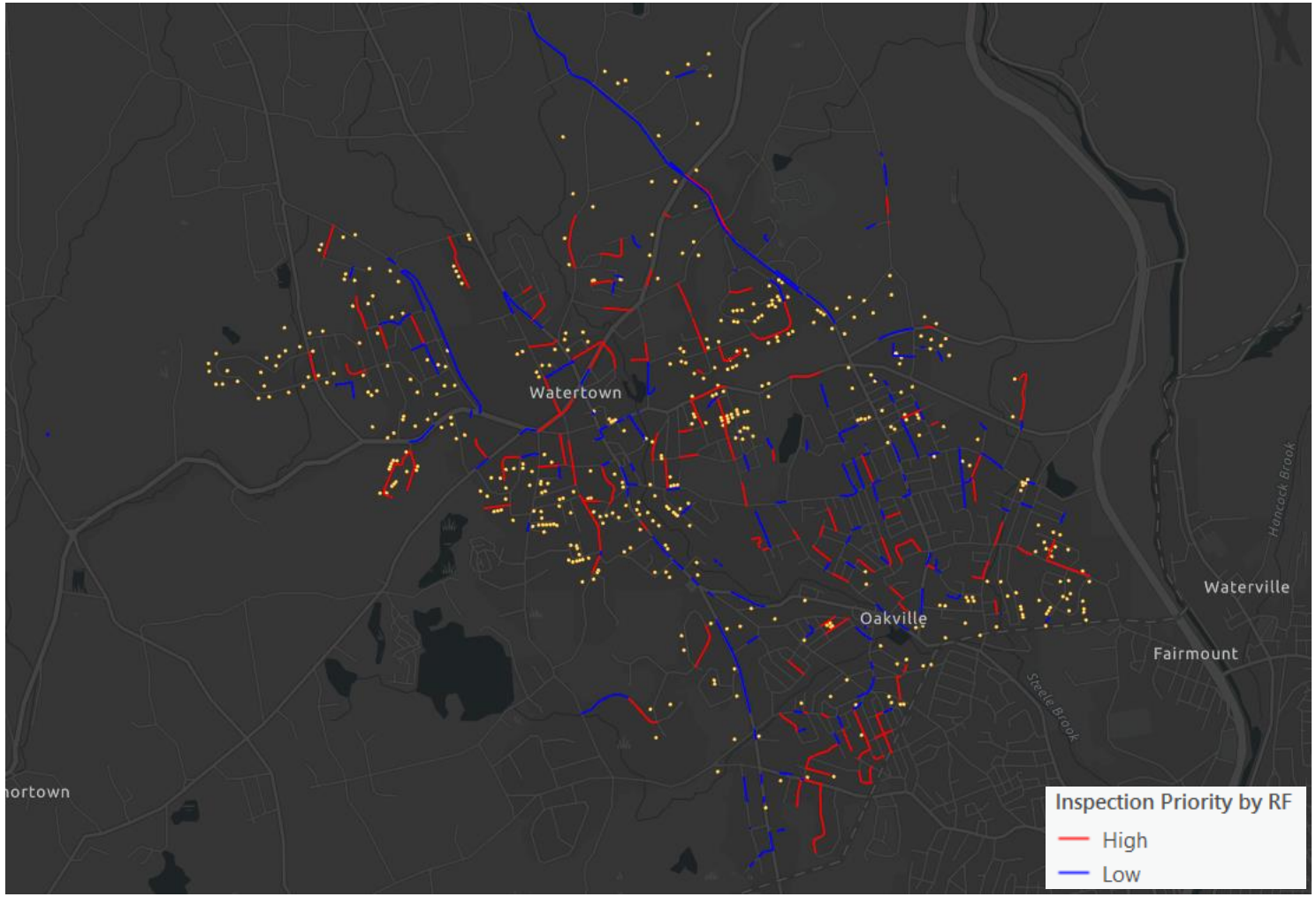
Model accuracy:
93.29%



Map: Decision Tree model's *Inspection Priority* predictions on 25% unseen raw data

Insp. Priority, RF on 25% Test Data

Model accuracy:
88.72%



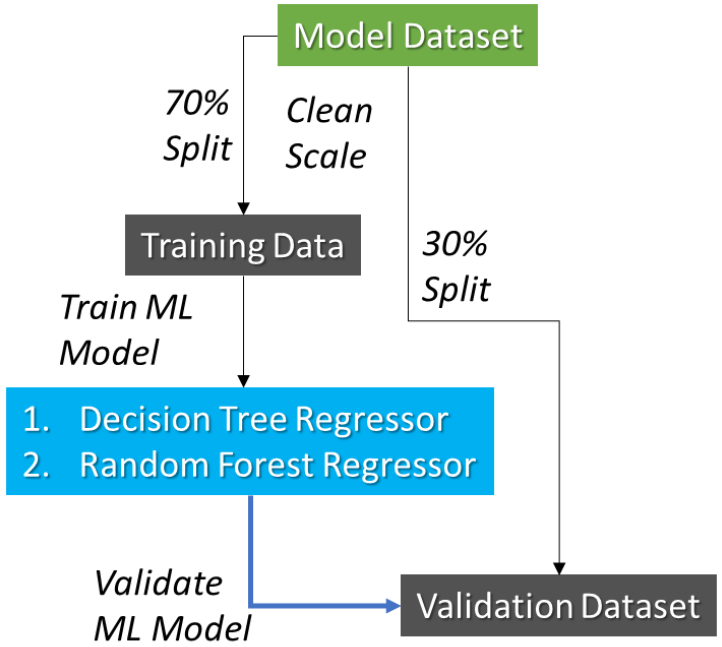
Map: Random Forest model's *Inspection Priority* predictions on 25% unseen raw data

Two Objectives

1. ML Classification for predicting “Inspection Priority”
2. ML Regression for predicting “Number of Breaks”

Training ML Regressors

- Decision Tree Regressor
- Random Forest Regressor

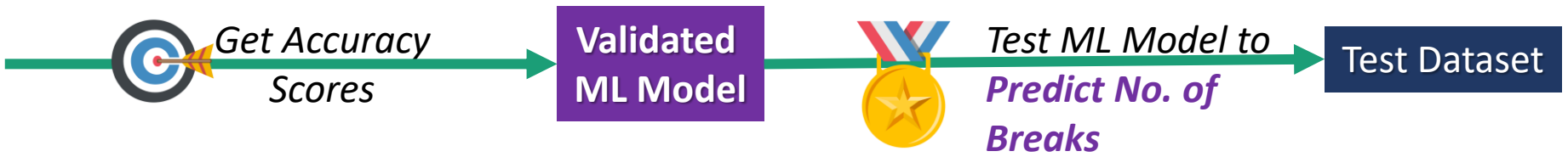



```
#1. Decision Tree Regression
#Creating decision Tree Regressor
from sklearn.tree import DecisionTreeRegressor
DT = DecisionTreeRegressor()
DT.fit(X_train, y_train)
y_pred_DT = DT.predict(X_test)
#Confusion Matrix
from sklearn.metrics import confusion_matrix, accuracy_score
Y_DT=y_pred_DT.astype(int)
print('Confusion Matrix for DT:')
conf_mat_DT = confusion_matrix(y_test, Y_DT)
print(conf_mat_DT)
print('Accuracy of DT Model:',accuracy_score(y_test, Y_DT)*100,'%')
print("Mean Absolute Error DT:",mean_absolute_error(y_test, Y_DT))
print("Mean Squared Error DT:", mean_squared_error(y_test, Y_DT))
```

```
#2. Random Forest Regression Model
# Creating a RF Regressor
from sklearn.ensemble import RandomForestRegressor
# creating a RF Regressor
RF = RandomForestRegressor(n_estimators = 100)
RF.fit(X_train,y_train)
y_pred_RF = RF.predict(X_test)
#Confusion Matrix
from sklearn.metrics import confusion_matrix, accuracy_score
Y_RF=y_pred_RF.astype(int)
print('Confusion Matrix for RF:')
conf_mat_RF = confusion_matrix(y_test, Y_RF)
print(conf_mat_RF)
print('Accuracy of RF Model:',accuracy_score(y_test, Y_RF)*100,'%')
print("Mean Absolute Error RF:",mean_absolute_error(y_test, Y_RF))
print("Mean Squared Error RF:", mean_squared_error(y_test, Y_RF))
```

Validation on *Validation Dataset*

	RANDOM FOREST REGRESSION	DECISION TREE REGRESSION
MEAN SQUARED ERROR	0.93	1.16
ACCURACY	78.90%	71.27%
CONFUSION MATRIX	[207 19 0 0 0 0 0] [22 8 1 0 0 0 0] [10 2 2 0 0 0 0] [1 0 0 0 0 0 0] [0 1 0 0 0 0 0] [1 0 0 0 0 0 0] [0 1 0 0 0 0 0]	[184 26 9 3 4 0 0] [15 12 1 1 0 2 0] [10 2 0 2 0 0 0] [1 0 0 0 0 0 0] [0 1 0 0 0 0 0] [0 0 0 0 0 0 0] [1 0 0 0 0 0 0]



Result: Test Dataset, No. of Breaks

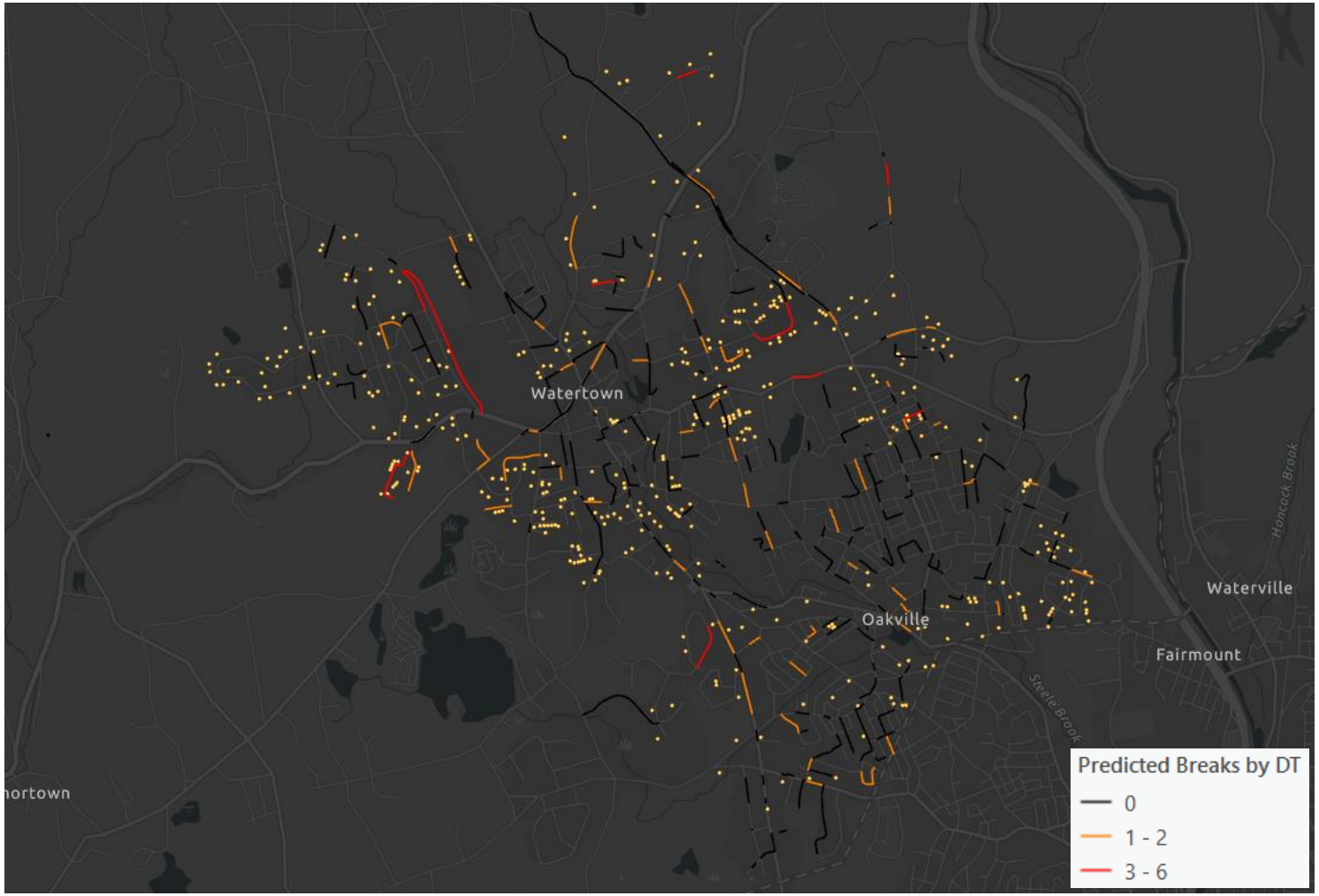
- ML Regressors applied to Test Dataset
- Number of Breaks predicted

ID	Label	Diameter	LENGTH_FT	Age	Ncustomers	PH	Pmax_Psi	Qmax_gpm	Acutal Breaks	Predicted Breaks by DT	Predicted Breaks by RF
29964	P-502	4	343	101	11	4.8	102	2	0	1	0
31123	P-308	8	1028	72	16	4.8	34	2	7	1	2
33476	P-3	6	456	101	24	4.8	98	0	0	1	0
33582	P-610	6	323	72	17	4.8	103	6	0	1	0
33803	P-369	4	188	101	13	4.8	95	1	0	1	0
33917	P-304	12	440	67	19	4.8	67	5	0	1	0
34086	P-151	6	1095	72	18	4.8	63	0	6	0	1
34134	P-625	6	651	101	30	4.8	102	22	0	1	0
34918	P-514	8	1449	72	34	4.8	106	8	2	1	2
35126	P-667	8	766	72	20	4.8	104	23	0	1	0
35327	P-857	12	840	101	7	4.8	107	160	0	1	0
35336	P-626	12	920	72	136	4.8	125	33	1	2	0
35449	P-1096	8	450	72	14	4.8	48	49	1	1	0
36191	P-566	6	545	112	9	4.8	70	6	0	1	0
36215	P-466	6	819	72	11	4.8	93	3	0	3	0
36223	P-105	12	215	67	4	4.8	102	0	0	1	0
36253	P-1045	8	878	101	13	4.8	76	153	0	0	1
49430	P-79	10	8394	44	0	4.8	165	898	2	0	1
31084	P-410	6	561	101	18	4.9	128	2	1	0	1
33701	P-216	8	160	101	17	4.9	69	1	0	1	0
35662	P-472	6	447	72	10	4.9	89	3	1	1	0
36290	P-89	12	1058	112	3	4.9	78	0	0	0	1
30612	P-357	6	438	72	15	5.2	120	2	0	1	0
30633	P-492	6	463	112	11	5.3	32	4	0	1	0
31142	P-582	6	1284	101	28	5.3	106	34	0	0	1
31287	P-418	6	641	67	10	5.3	120	2	0	2	1
31358	P-482	8	612	101	74	5.3	79	7	1	1	0
31754	P-1010	6	849	101	16	5.3	55	72	0	1	0

Random Samples from Test Dataset

of Breaks, DT on 25% Test Data

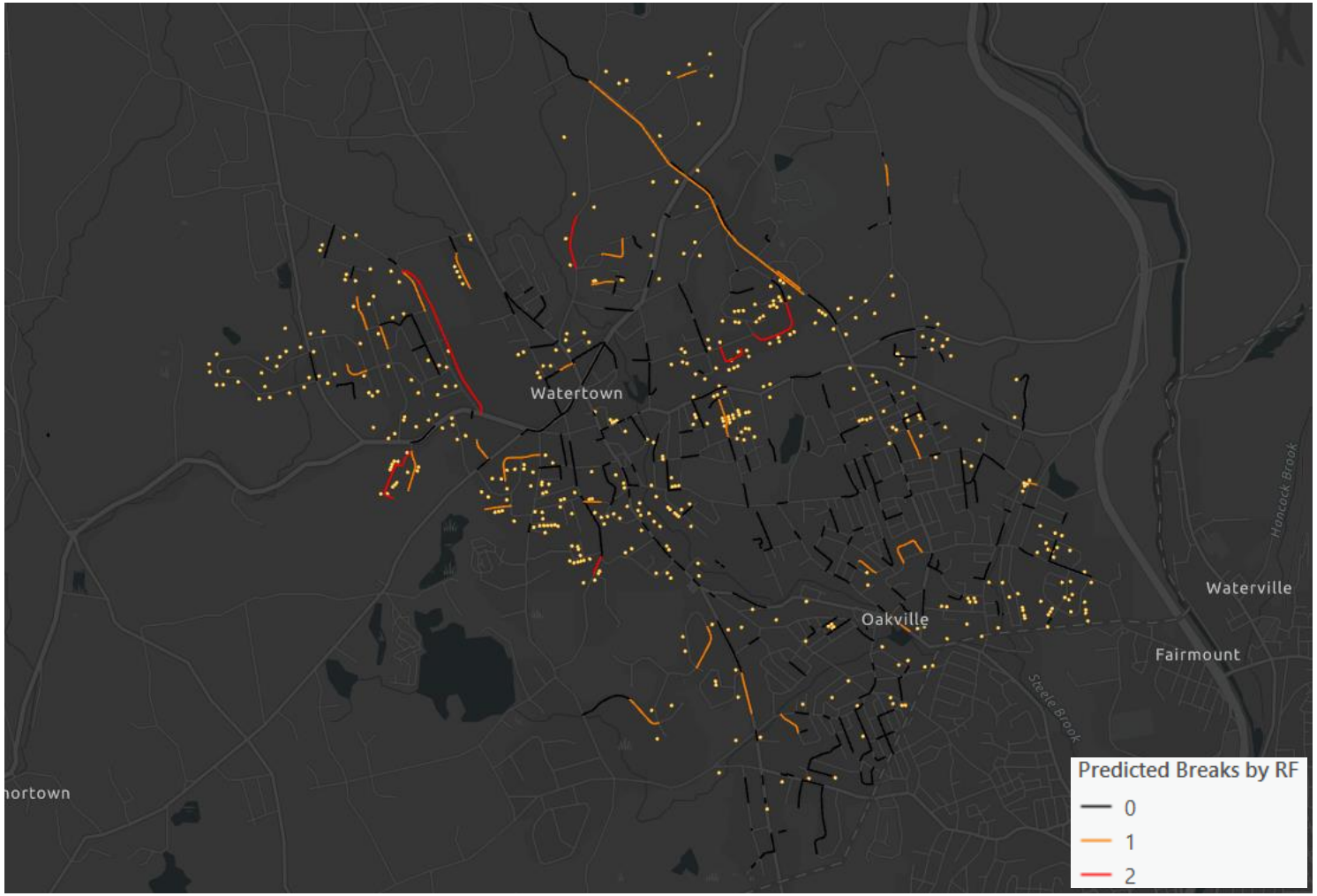
Model accuracy:
71.27%



Map: Decision Tree model's *# of Breaks* predictions on 25% unseen raw data

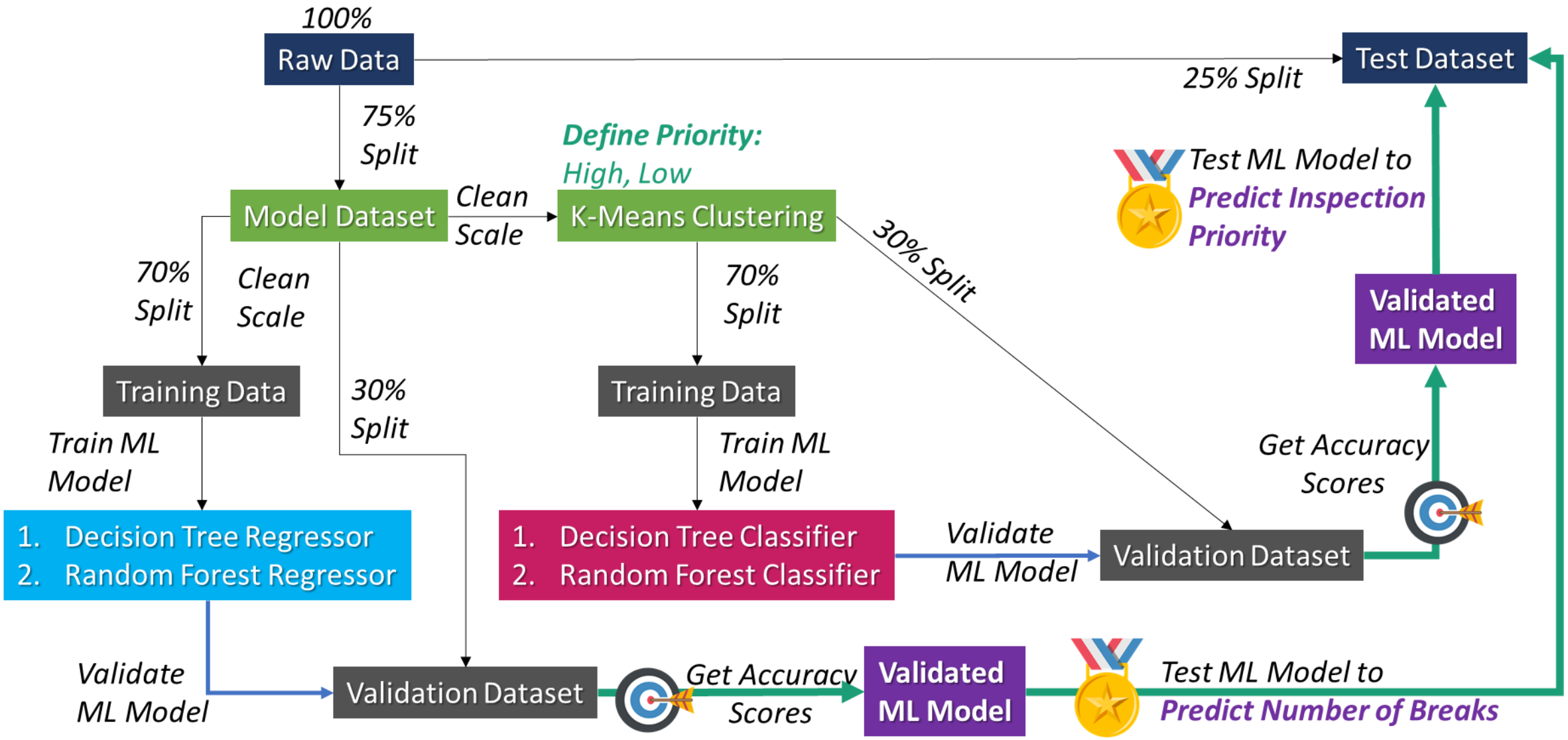
of Breaks, RF on 25% Test Data

Model accuracy:
78.90%



Map: Random Forest model's *# of Breaks* predictions on 25% unseen raw data

Summary


100%
Raw Data
Test Dataset
25% Split
75%
Split
Define Priority:
High, Low
Model Dataset
Clean
Scale
K-Means Clustering
Test ML Model to
Predict Inspection
Priority
70%
Split
Clean
Scale
30% Split
70%
Split
Validated
ML Model
Training Data
30%
Split
Training Data
Train ML
Model
Train ML
Model
Get Accuracy
Scores
1. Decision Tree Regressor
2. Random Forest Regressor
1. Decision Tree Classifier
2. Random Forest Classifier
Validate
ML Model
Validation Dataset
Validate
ML Model
Validation Dataset
Get Accuracy
Scores
Validated
ML Model
Test ML Model to
Predict Number of Breaks

Conclusions

- Machine Learning has a huge application scope in operations and maintenance of water infrastructure systems
- Predicting Inspection Priority:
  - Decision Tree (*93.29%*) performed better than Random Forest (*88.73%*) to handle the *Classification problem*
- Predicting Number of Breaks:
  - Random Forest (*78.90%*) performed better than Decision Tree (*71.27%*) to handle the *Regression problem*

Future Scope

- Planning of maintenance activities using predictions and asset-to-asset relation.
  - Considering Water Infrastructure with Transportation Infrastructure
- Understand influence of dynamic features such as pressure, discharge, age, etc. that change over time to build a prediction model

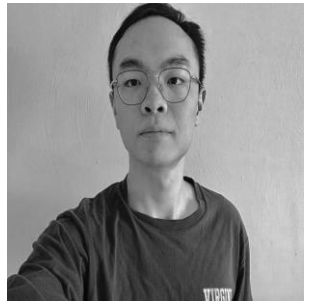
12-750 INFRASTRUCTURE MANAGEMENT PROJECT
SPRING 2022

Predicting Inspection Priority and Pipe Breaks in Water Distribution Network Using ML

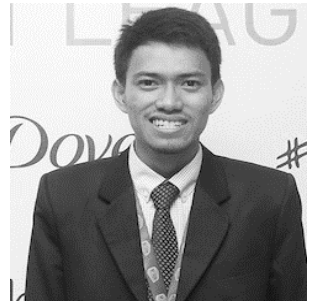
A Statistical 'Learning and Modeling' Study for Infrastructure Management

Devashri Karve
Graduate Student
dkarve@andrew.cmu.edu
CEE Department
Carnegie Mellon University

Tanay Kulkarni
Graduate Student
tskulkar@andrew.cmu.edu
CEE Department
Carnegie Mellon University



Yijie Zhu
Graduate Student
yijiezhu@andrew.cmu.edu
CEE Department
Carnegie Mellon University



Zulkifli Palinrungi
Graduate Student
zpalinru@andrew.cmu.edu
Energy Science,
Technology, and Policy
Carnegie Mellon University

Mentor
Prof. Dr. Donald Coffelt