Road Mapping, Initiative Definition and Project Planning

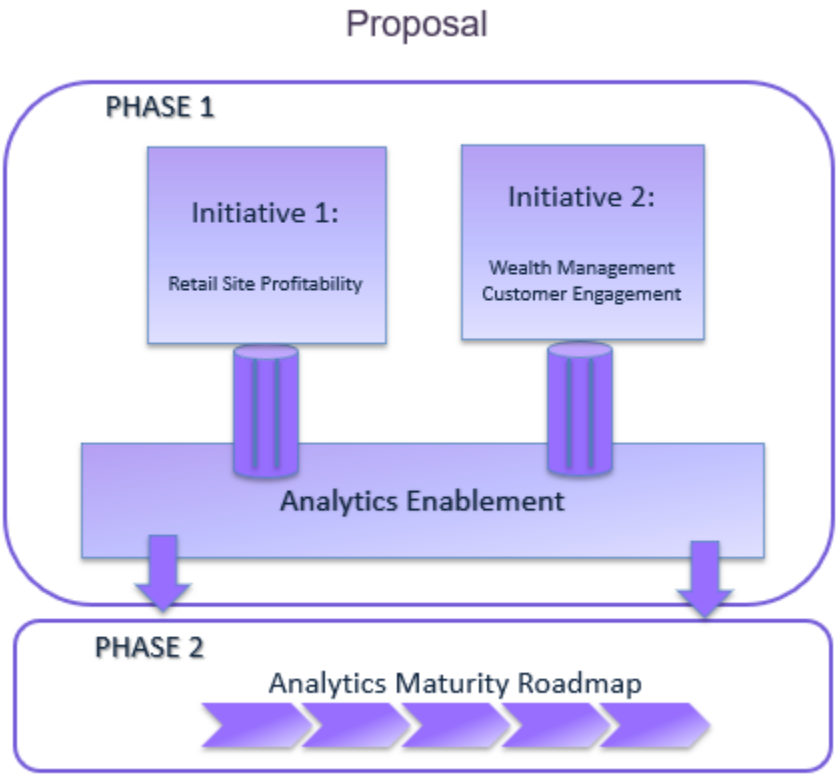
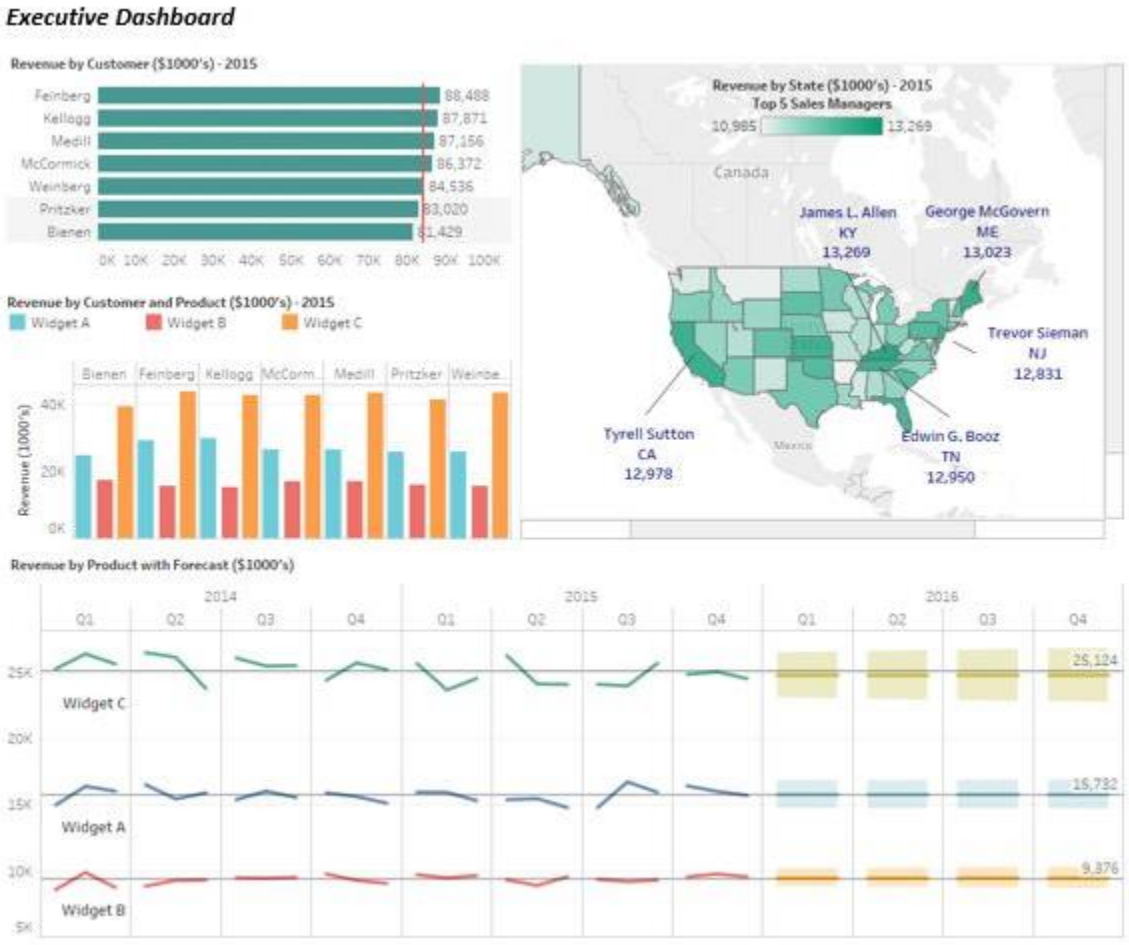
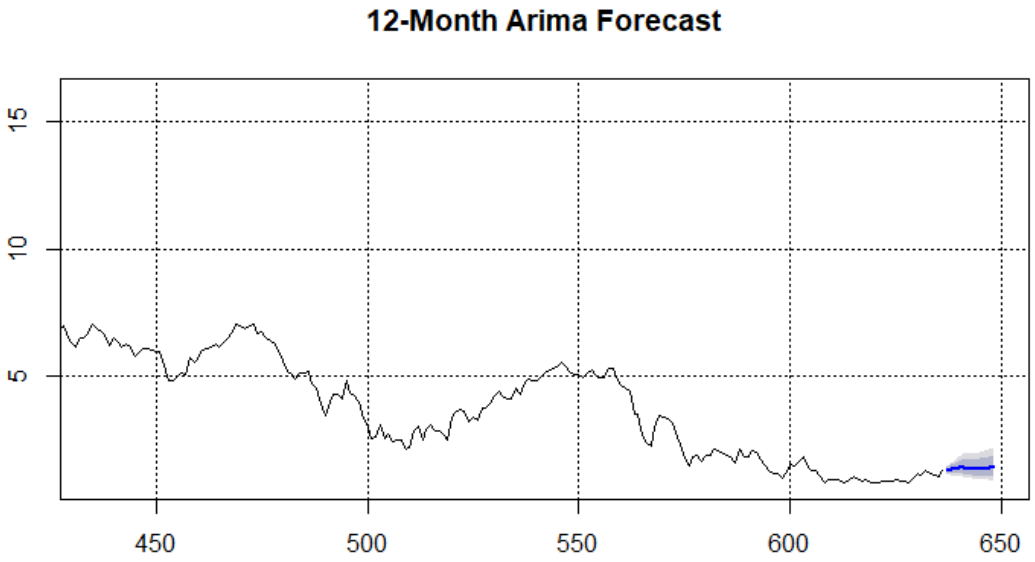


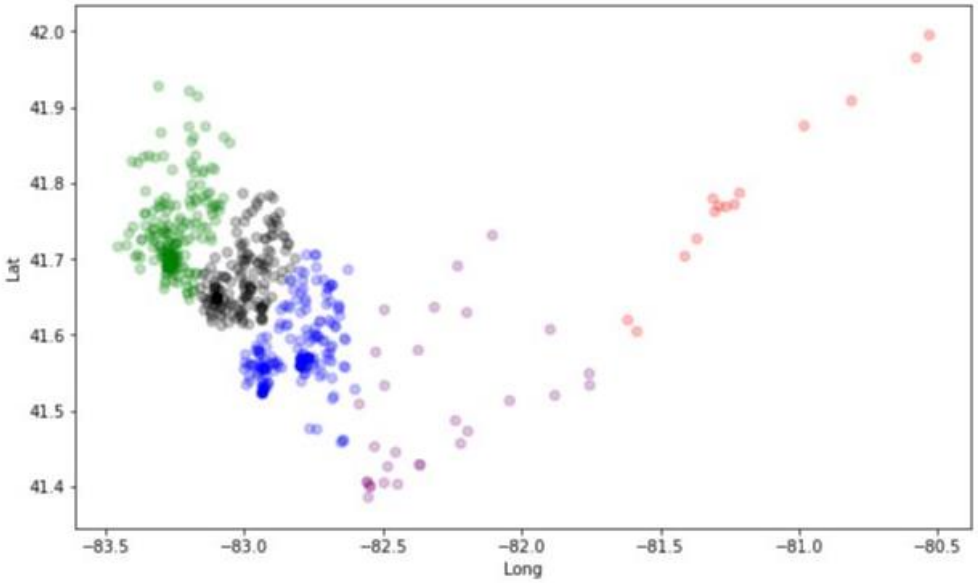
Tableau Dashboard & KPIs



Forecasting



Clustering



Exploratory Data Analysis

```
safe = data.loc[data['Microcystin, Total'].between(0.0, 0.29),['Microcystin, Total']]
infant = data.loc[data['Microcystin, Total'].between(0.3, 0.99),['Microcystin, Total']]
drink = data.loc[data['Microcystin, Total'].between(1, 1.59),['Microcystin, Total']]
adult = data.loc[data['Microcystin, Total'].between(1.6, 5.9),['Microcystin, Total']]
PHA = data.loc[data['Microcystin, Total'].between(6.0, 19.9),['Microcystin, Total']]
NCA = data.loc[data['Microcystin, Total']>=20,['Microcystin, Total']]
```

```
safe_percent = round((safe.shape[0]/data.shape[0]),2)
infant_percent = round((infant.shape[0]/data.shape[0]),2)
drink_percent = round((drink.shape[0]/data.shape[0]),2)
adult_percent = round((adult.shape[0]/ data.shape[0]),2)
PHA_percent = round((PHA.shape[0]/ data.shape[0]),2)
NCA_percent = round((NCA.shape[0]/data.shape[0]),2)
```

```
print ('\nsafe %', (safe_percent*100))
print ('\ninfant %', (infant_percent*100))
print ('\ndrink %', (drink_percent*100))
print ('\nadult %', (adult_percent*100))
print ('\nPHA %', (PHA_percent*100))
print ('\nNCA %', (NCA_percent*100))
```

```
safe % 31.0

infant % 50.0

drink % 6.0

adult % 10.0

PHA % 2.0

NCA % 0.0
```

Decision Analysis / Optimization

Decision Variables:

Units	Grand Rapids	Lansing	Phoenix	Baltimore	Phoenix Site Indicator	Baltimore Site Indicator	M
CA	X_{11}	X_{12}	X_{13}	X_{14}	y_3	y_4	M_1
NY	X_{21}	X_{22}	X_{23}	X_{24}	y_3	y_4	M_2

$y_3, y_4 \in \{0,1\}$

$M_1, M_2 = 10{,}000$

Objective: Minimize Cost: Z = 8x11 + 4x12 + 5x13 + 4x14 + 5x21(y3) + 7x22(y4) + 6x23(y3) + 6x24(y4)

Constraints:

Supply/Demand Retail

CA: $x_{11} + x_{12} + x_{13} + x_{14} >= 900$

NY: $x_{21} + x_{22} + x_{23} + x_{24} >= 1100$

Plant Supply/Capacity

Grand Rapids: $x_{11} + x_{21} <= 800$

Lansing: $x_{12} + x_{22} <= 700$

Phoenix: $x_{13} + x_{23} <= 400$

Baltimore: $x_{14} + x_{24} <= 400$

Phoenix or Baltimore

$x_{13} + x_{23} - M_1(y_3) <= 0$

$x_{14} + x_{24} - M_2(y_4) <= 0$

$y_3 + y_4 <= 1$

Non-negative

$x_{11}, x_{12}, x_{13}, x_{14}, x_{21}, x_{22}, x_{23}, x_{24} >= 0$

Integers

$x_{11}, x_{12}, x_{13}, x_{14}, x_{21}, x_{22}, x_{23}, x_{24}$ = int

Binary

y_3, y_4 = bin

In [23]:
```
#Training model by using neural network
with tf.Session() as sess:
    init.run()
    for epoch in range(n_epochs):
        for X_batch, y_batch in shuffle_batch(X_train, y_train, batch_size):
            sess.run(training_op, feed_dict={X: X_batch, y: y_batch})
        acc_batch_M2 = accuracy.eval(feed_dict={X: X_batch, y: y_batch})
        acc_val_M2 = accuracy.eval(feed_dict={X: X_valid, y: y_valid})
        print(epoch, "Batch accuracy:", acc_batch_M2, "Val accuracy:", acc_val_M2)

    save_path = saver.save(sess, "./my_model_final.ckpt")
```

```
0 Batch accuracy: 0.9 Val accuracy: 0.9028
1 Batch accuracy: 0.92 Val accuracy: 0.9302
2 Batch accuracy: 0.96 Val accuracy: 0.9448
3 Batch accuracy: 0.92 Val accuracy: 0.9512
4 Batch accuracy: 0.96 Val accuracy: 0.9586
5 Batch accuracy: 0.94 Val accuracy: 0.9566
6 Batch accuracy: 1.0 Val accuracy: 0.9634
7 Batch accuracy: 0.98 Val accuracy: 0.9682
8 Batch accuracy: 1.0 Val accuracy: 0.9702
9 Batch accuracy: 0.98 Val accuracy: 0.9668
10 Batch accuracy: 0.98 Val accuracy: 0.9718
11 Batch accuracy: 1.0 Val accuracy: 0.9742
12 Batch accuracy: 0.98 Val accuracy: 0.9714
13 Batch accuracy: 1.0 Val accuracy: 0.9696
14 Batch accuracy: 1.0 Val accuracy: 0.9758
15 Batch accuracy: 0.96 Val accuracy: 0.9746
16 Batch accuracy: 1.0 Val accuracy: 0.9754
17 Batch accuracy: 1.0 Val accuracy: 0.9754
18 Batch accuracy: 1.0 Val accuracy: 0.978
19 Batch accuracy: 1.0 Val accuracy: 0.9774
```