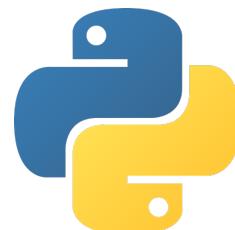


# Highlight of Major Projects

Skills learned:

- ✓ Git
- ✓ Python / Data Structures
- ✓ SQL
- ✓ Supply Chain Forecasting
- ✓ Data Visualization with Power BI
- ✓ Service-oriented IT
- ✓ Machine Learning



- Memory Usage
  - Tracking usage of supply chain planning tool (OMP CHS)
  - Outcomes:
    - Daily tracking of issues
    - Proactive actions
- Equipment Forecasting for Eastman
  - Outcomes:
    - A working DRP Flow forecast with equipment needed and shipping conditions

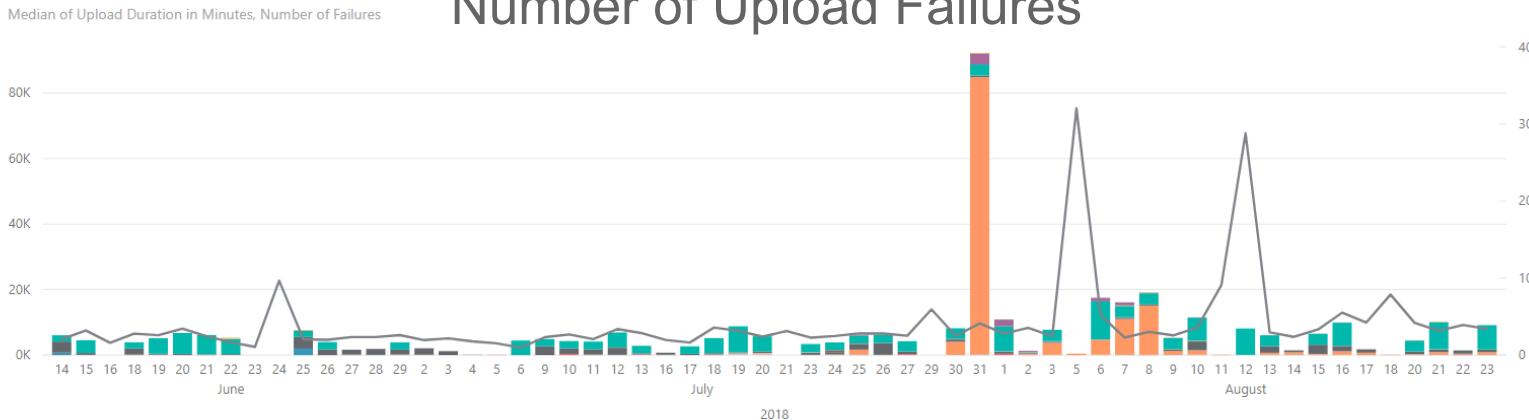


# OMP Upload Durations



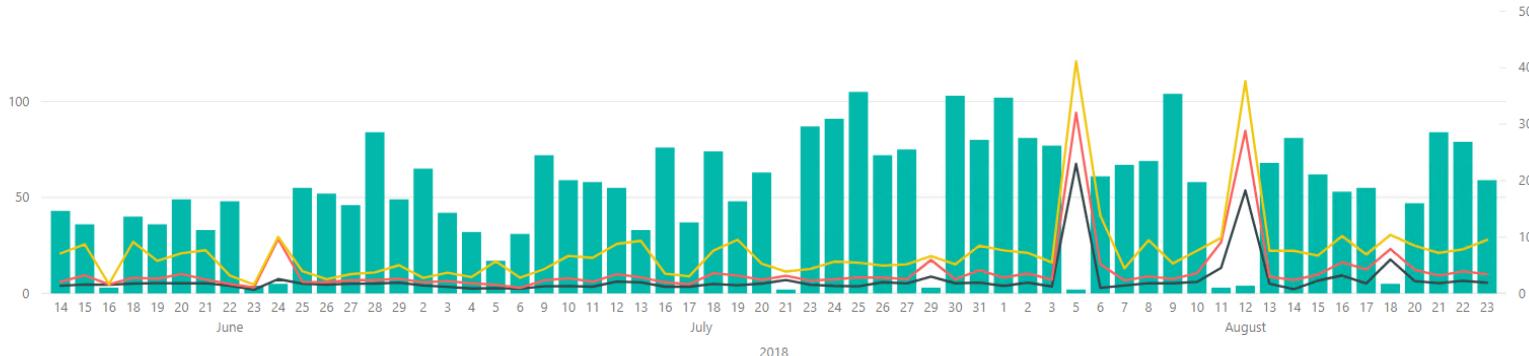
# OMP Upload Durations

## Number of Upload Failures



Max of Upload Duration in Hrs. by Year, Month and Day

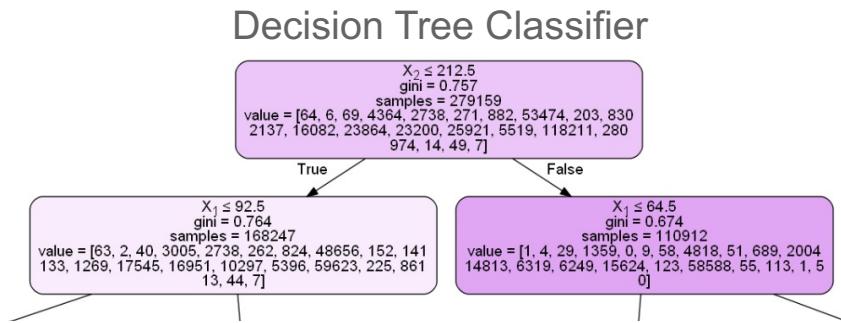
● Nb Uploads ● Upload Duration 25th Pctile (Min) ● Upload Duration 50th Pctile (Min) ● Upload Duration 75th Pctile (Min)



# Machine Learning Model

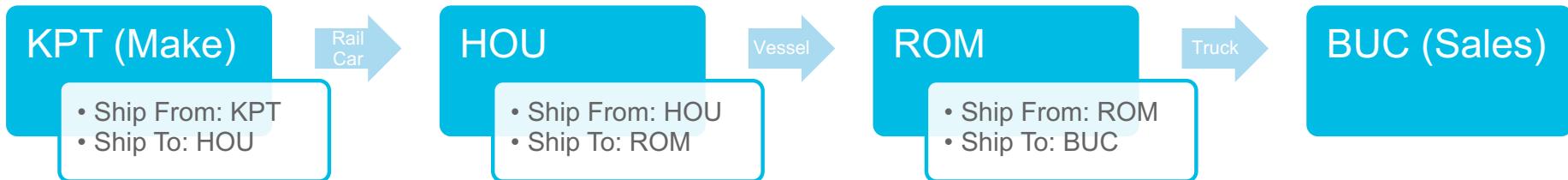
*Data Forecasting*

# A Typical Flow Forecast



## Forecast Predictor:

1. Calculates DRP (Distributions Requirement Planning)
2. Derives shipping condition and equipment needed to ship to sales



# Predicting using a Flow Forecast

Training RFC

Given:  
Delivery history

Train RFC model

Shipping  
condition +  
equipment  
needed

Applying RFC

Given:  
Forecast

DRP Flows

Predicted shipping  
condition +  
equipment needed

# Equipment Forecasting

- BAxTR – Basic ApproXimator for Transportation Resources
  - Used delivery history data to train machine learning models
    - Out of 3 approaches, highest accuracy is 95% by using Random Forest Classifier

```
127     dump(rfc_low_lvl, 'rfc_low_lvl_final.joblib')
128     dump(dtc_low_lvl, 'dtc_low_lvl_final.joblib')
129     dump(ml.le_y_scnd, 'ShipCond_encoder.joblib')
130     #rfc_low_lvl_repo = load('rfc_low_lvl.joblib')
131     #dtc_low_lvl_repo = load('dtc_low_lvl.joblib')
132

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

C:\Repos\BAxTR>C:/Users/U778411/AppData/Local/Programs/Python/Python37/python.exe c:/Repos/BAxTR/baxtr_predictive_main.py
C:\Users\U778411\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\externals\six.py:31: DeprecationWarning:
e official version of six (https://pypi.org/project/six/), \(https://pypi.org/project/six/\), DeprecationWarning)
C:\Users\U778411\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning: The
"10 in version 0.20 to 100 in 0.22.", FutureWarning)

Final:
Accuracy for RFC: 95.58 % ← 95.58%
Accuracy for DTC: 95.57 %
C:\Users\U778411\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning: The
"10 in version 0.20 to 100 in 0.22.", FutureWarning)

v1:
Accuracy for RFC: 87.48 %
Accuracy for DTC: 87.48 %
C:\Users\U778411\AppData\Local\Programs\Python\Python37\lib\site-packages\sklearn\ensemble\forest.py:245: FutureWarning: The
"10 in version 0.20 to 100 in 0.22.", FutureWarning)

v2:
Accuracy for RFC: 88.51 %
Accuracy for DTC: 88.52 %

C:\Repos\BAxTR>
```

# Other notable term Projects

- Learning SQL for querying data
- Machine learning in Python
- Service-oriented IT

The screenshot shows a Visual Studio Code interface with a dark theme. On the left is a code editor window titled "MachineLearning.py" containing Python code for machine learning. The code imports numpy, matplotlib, and sklearn. It defines a "model" function that generates data with two oscillations and noise, and fits a Random Forest Regressor to it. A plot titled "Figure 1" shows the data points with error bars and a fitted curve. The bottom of the screen shows the VS Code navigation bar with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, and TERMINAL.

```
mininal Help MachineLearning.py - Python Practice - Visual Studio Code

MachineLearning.py x
MachineLearning.py > ...
43 # Example of bagging
44 tree = DecisionTreeClassifier()
45 bag = BaggingClassifier(tree, n_estimators = 100, max_samples = 0.8, random_state = 1)
46 bag.fit(X, y)
47 visualize_classifier(bag, X, y)
48 plt.show()
49
50 modelForClassifier = RandomForestClassifier(n_estimators = 100, random_state = 0)
51 visualize_classifier(modelForClassifier, X, y)
52 plt.show()
53
54
55 # Random Forest Regression
56
57 rng = np.random.RandomState(42)
58 x = 10 * rng.rand(200)
59
60 def model(x, sigma = 0.3):
61     fast_oscillation = np.sin(5 * x)
62     slow_oscillation = np.sin(0.5 * x)
63     noise = sigma * rng.randn(len(x))
64
65     return slow_oscillation + fast_oscillation + noise
66
67 y = model(x)
68 plt.errorbar(x, y, 0.3, fmt = 'o')
69 plt.show()
70
71 from sklearn.ensemble import RandomForestRegressor
72 forest = RandomForestRegressor(200)
73 forest.fit(x[:, None], y)
74
75 xfit = np.linspace(0, 10, 1000)
76 yfit = forest.predict(xfit[:, None])
77 ytrue = model(xfit, sigma = 0)
78
79 plt.errorbar(x, y, 0.3, fmt = 'o', alpha = 0.5)
80 plt.plot(xfit, yfit, '-r')
81 plt.plot(xfit, ytrue, '-k', alpha = 0.5)
82 plt.show()

PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL
xfit = np.linspace(0, 10, 1000)
AttributeError: module 'numpy' has no attribute 'linspace'
2: Python
```

# Project Next Steps

- Continue gathering OMP server status and upload times
- Continue predictions for flow forecast



```
# baxr_predictive_main.py      # baxr_predict_ship_cond.py      # baxr_machine_learning.py

18 rfc_low_lvl_v1 = load('rfc_low_lvl_v1.joblib')
19 dtc_low_lvl_v1 = load('dtc_low_lvl_v1.joblib')
20 rfc_low_lvl_v2 = load('rfc_low_lvl_v2.joblib')
21 dtc_low_lvl_v2 = load('dtc_low_lvl_v2.joblib')
22
23 # Load the ShipCond encoder that was used for training sets
24 le_y_scnd = load('ShipCond_encoder.joblib')
25
26 # Filter csv for only "Move" SrcType
27 df_flow_forecast = df_flow_forecast[df_flow_forecast['SrcType'] == 'Move']
28 # Filter csv for X_ShipToEncoded that is not empty
29 df_flow_forecast = df_flow_forecast[df_flow_forecast['X_ShipToEncoded'].notnull()]
30
31 # Select the encoded data columns
32 df_flow_forecast_ml = df_flow_forecast[['X_FlowTypeEncoded', 'X_ShipFromEncoded', 'X_ShipToEncoded',
33                                         'X_ProdOpEncoded', 'X_TrspOpEncoded', 'X_ShipFromCtryEncoded',
34                                         'X_ShipFromCtryCategEncoded', 'X_ShipToCtryEncoded',
35                                         'X_ShipToCtryCategEncoded']].copy()
36
36 X_low_lvl_final = df_flow_forecast_ml.values[:, (0,1,2,3,4,5,7)]
37 X_low_lvl_v1 = df_flow_forecast_ml.values[:, (0,6,8,9,3)]
38 X_low_lvl_v2 = df_flow_forecast_ml.values[:, (0,1,8,9,3)]
39
40 # Run RFC and DTC models on the selected columns
41 y_rfc_pred_low_lvl_final = rfc_low_lvl_final.predict(X_low_lvl_final)
42 y_dtc_pred_low_lvl_final = dtc_low_lvl_final.predict(X_low_lvl_final)
43 y_rfc_pred_low_lvl_v1 = rfc_low_lvl_v1.predict(X_low_lvl_v1)
44 y_dtc_pred_low_lvl_v1 = dtc_low_lvl_v1.predict(X_low_lvl_v1)
45 y_rfc_pred_low_lvl_v2 = rfc_low_lvl_v2.predict(X_low_lvl_v2)
46 y_dtc_pred_low_lvl_v2 = dtc_low_lvl_v2.predict(X_low_lvl_v2)
47
48 # inverse_transform the encoded data to get readable predicted ShipCond
49 df_flow_forecast_benchmark = df_flow_forecast.copy()
50
51 # Create and store new columns for RFC and DTC predictions, save to new csv file
52 rfc_low_lvl_v1_pred = le_y_scnd.inverse_transform(y_rfc_pred_low_lvl_v1)
53 df_flow_forecast_benchmark['RFC_Low_Lvl_Prediction'] = rfc_low_lvl_v1_pred
54 dtc_low_lvl_v1_pred = le_y_scnd.inverse_transform(y_dtc_pred_low_lvl_v1)
55 df_flow_forecast_benchmark['DTC_Low_Lvl_Prediction'] = dtc_low_lvl_v1_pred
56 df_flow_forecast_benchmark.to_csv('final_drp_forecast_before_aim_v1_with_final_predictions.csv', index=False)
57
58 rfc_low_lvl_v1_pred = le_y_scnd.inverse_transform(y_rfc_pred_low_lvl_v1)
59 df_flow_forecast_benchmark['RFC_Low_Lvl_Prediction'] = rfc_low_lvl_v1_pred
60 dtc_low_lvl_v1_pred = le_y_scnd.inverse_transform(y_dtc_pred_low_lvl_v1)
61 df_flow_forecast_benchmark['DTC_Low_Lvl_Prediction'] = dtc_low_lvl_v1_pred
62 df_flow_forecast_benchmark.to_csv('final_drp_forecast_before_aim_v1_with_v1_predictions.csv', index=False)
63
64 rfc_low_lvl_v1_pred = le_y_scnd.inverse_transform(y_rfc_pred_low_lvl_v1)
65 df_flow_forecast_benchmark['RFC_Low_Lvl_Prediction'] = rfc_low_lvl_v1_pred
66 dtc_low_lvl_v1_pred = le_y_scnd.inverse_transform(y_dtc_pred_low_lvl_v1)
67 df_flow_forecast_benchmark['DTC_Low_Lvl_Prediction'] = dtc_low_lvl_v1_pred
68 df_flow_forecast_benchmark.to_csv('final_drp_forecast_before_aim_v1_with_v2_predictions.csv', index=False)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL  
"10 in version 0.28 to 100 in 0.22.", FutureWarning)