Modelling Summary

Pycaret – Does rainfall information (intensity or total depth) matter when making influent flow predictions? Can we have a model that can predict the influent flow on a future storm?

1. Developed model based on individual storms (June 2018 and April 2018)
2. trained, tested and improved model performance (minimize errors in predictions)
3. Made predictions on a future storm
4. Quality of model output depended on the range of inputs (water quality, flow control and precipitation)
5. First try showed that flow control is the best predictor (June 2018 storm to predict the October 2019 storm). over prediction that <10% in error
6. Second try excluded flow control and considered on water quality and precipitation – precipitation ranked low again on the importance of influent prediction and model prediction accuracy reduced (April 2018 storm to predict July 2019 storm) i.e. under prediction of ~38% for flows above 350MGD and over prediction of ~60% for flows below 350MGD.
7. Anomaly detection – finding patterns within data that doesn’t conform to a specific behaviour (translate usually into actionable information - leading to new features that might be introduced in the model or lead to new findings). Outlier is just a rare chance of occurrence (observation point is distant from other observation points and usually not fully explainable – improve model accuracy).

**Way Forward**

1. too few attributes lead to information loss and too many attributes lead to computational complexity – select proper feature dimension before model training (principal component analysis??) Likely going to do some dimension reduction. Some new attributes will be created by combinining the original variables but this will be in a way to enable new attributes reflect information in the original variables as much as possible and variables are not mutually correlated.

**Pros - running time of model will reduce and effective attributes are annihilate (pronounced anailated) by the curse of dimensionality.**

bagging – bootstrap aggregation is when subsamples of a population are used to estimate population parameters when statistical distribution cannot be figured out a priori. more variability is introduced to provide a more robust estimate of population parameters through random resampling with replacement.

random forest model is easy to use, robustness towards messy data and parallelizability.

to-do list

use machine learning to predict influent flow (include rainfall data) to show prediction accuracy.

if a model is able to make accurate predictions on unseen data then the model is able to generalize training data to the test data. ideally, we want to generalize the model as accurately as possible. building a model that is too complex for the information available is **overfitting**. Not able to generalize well in the new data. building too simple a model is underfitting i.e. not able to capture all the aspects and variability in the data (model does badly on the training dataset)

include feature engineering – interactions between inputs

discrepancy between training and testing R2 is a sign of overfitting.

looking for a less complex model that perform worse during training, but better generalization.

number of features considered in the model??

why linear models are good - linear models are fast to train and also fast to predict and scale to very large datasets?? and easy to understand how the prediction is made.

random forest models are slower to train and predict

lasso and ridge regression models restrict the coefficients close to zero and fit an additional constraint.

under estimation of the prediction might be due to small sample size containing extreme values.

look into the development of partial dependence plots