EY Data Challenge Phase 2

Benjamin Nicholson & Jonah Zembower

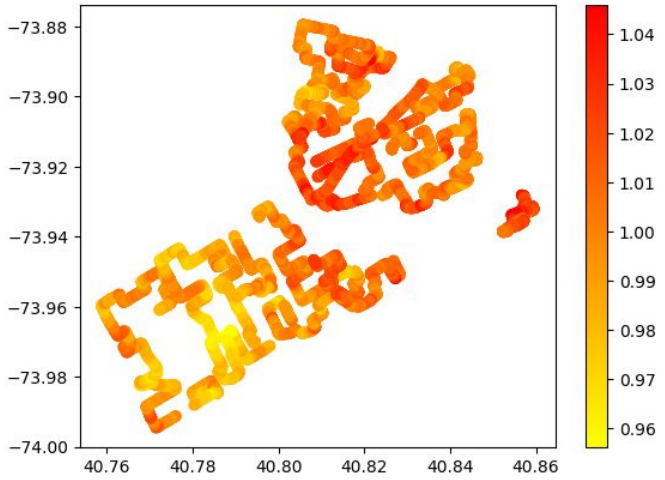
Recap from last meeting

Project Goal: We are looking to predict changes in temperatures within NYC, name ‘Urban Heat Islands’

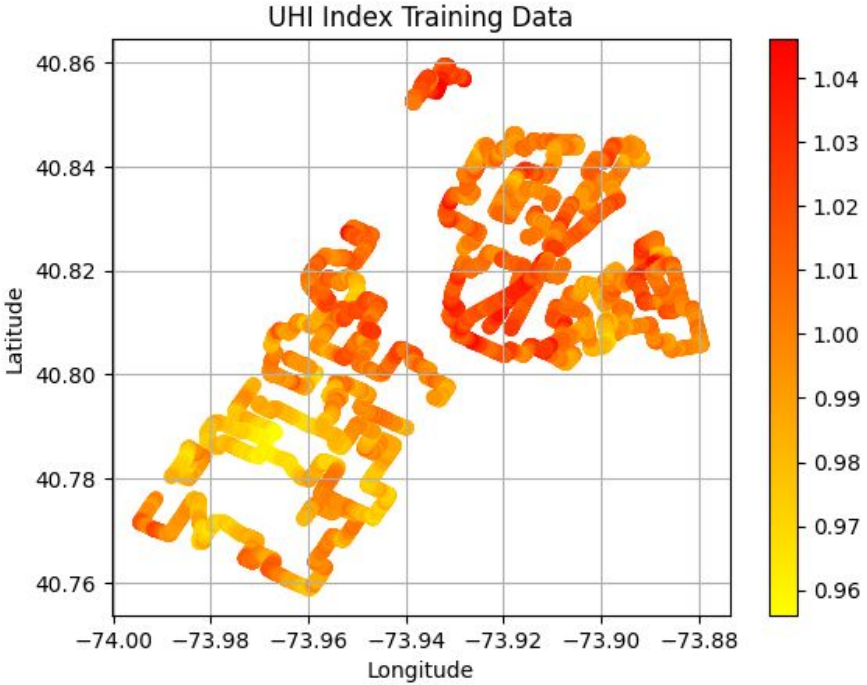
Review of data: Access to satellite data which can be used in building models. Considering looking into NYC Open Data to include more features

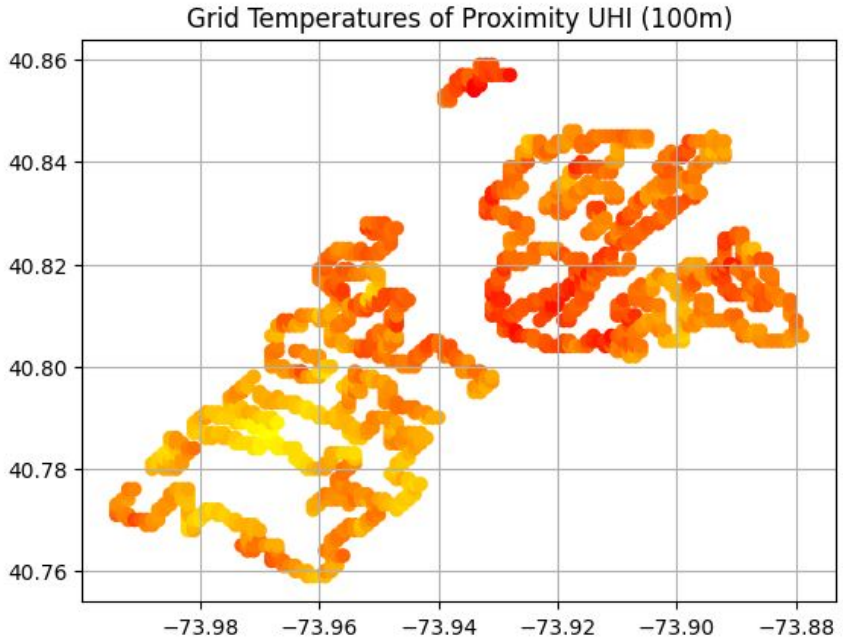
Next Steps: Average income, heat conductivity of materials, weather data, use grid instead of long/lat

	Longitude	Latitude	datetime	UHI Index
0	-73.919037	40.814292	24-07-2021 15:53	1.034616
1	-73.918978	40.814365	24-07-2021 15:53	1.028125
2	-73.918927	40.814433	24-07-2021 15:53	1.028125
3	-73.918875	40.814500	24-07-2021 15:53	1.025961
4	-73.918827	40.814560	24-07-2021 15:53	1.025961



Granular (6 d.p.) vs Proximity Approach (3 d.p) for UHI Index

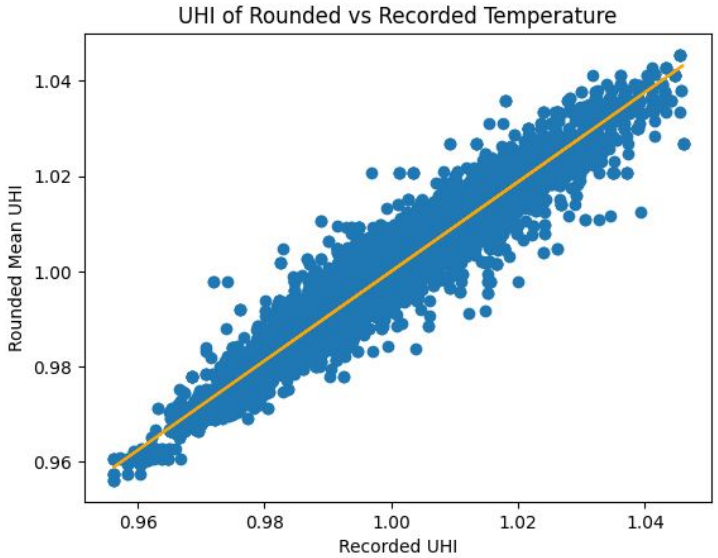




Two options for Model training (we are yet to do the proximity approach for UHI Values)

1: Proximity Training

R^2 = 0.94



2. Granular Training

We have done all of our models so far on granular training as this has been a new idea but we have been investigating different ways of doing.

cKDTrees is a data structure which has allowed us to map nearby points even if they are not the same long / lat (down to multiple decimal places)

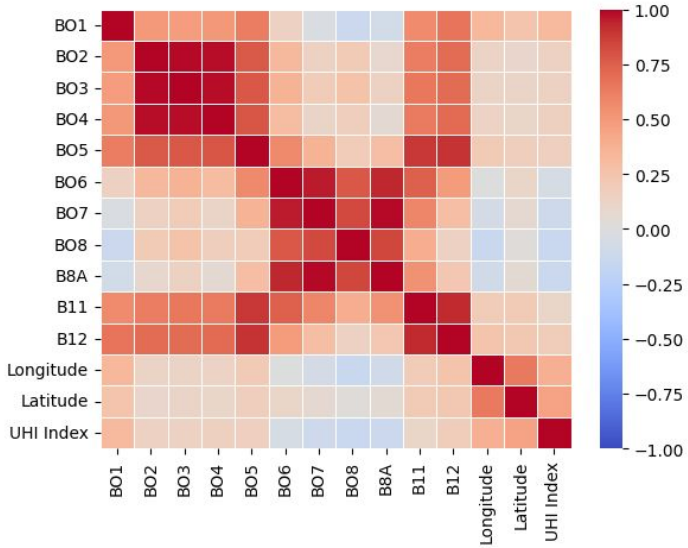
Data Available: Sentinel 2

Band combinations we will look to build into models:

Vegetation and Plant Health: NDVI = (B8 - B4) / (B8 + B4)

Vegetation and Water: NDWI: NDWI = (B3 - B8) / (B3 + B8)

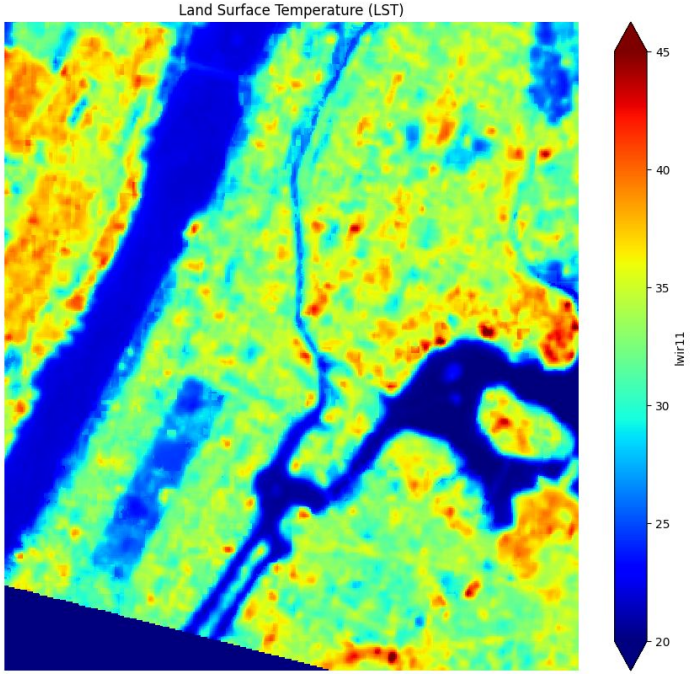
Urban and Built Up: NDBI = (B11 - B8) / (B11 + B8)



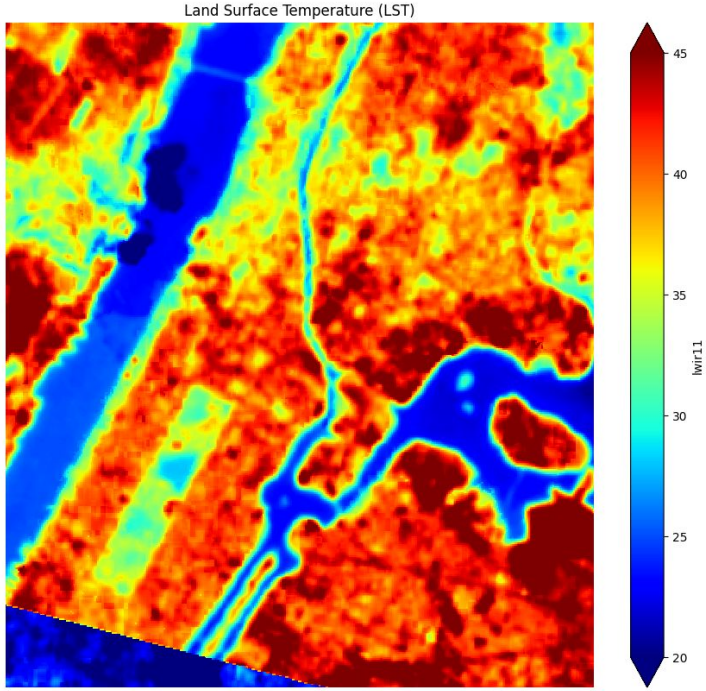
Sentinel-2 Bands	Central Wavelength (µm)	Resolution (m)
Band 1 - Coastal aerosol	0.443	60
Band 2 - Blue	0.490	10
Band 3 - Green	0.560	10
Band 4 - Red	0.665	10
Band 5 - Vegetation Red Edge	0.705	20
Band 6 - Vegetation Red Edge	0.740	20
Band 7 - Vegetation Red Edge	0.783	20
Band 8 - NIR	0.842	10
Band 8A - Vegetation Red Edge	0.865	20
Band 9 - Water vapour	0.945	60
Band 10 - SWIR - Cirrus	1.375	60
Band 11 - SWIR	1.610	20
Band 12 - SWIR	2.190	20

Data Available: Landsat Temperature

2021

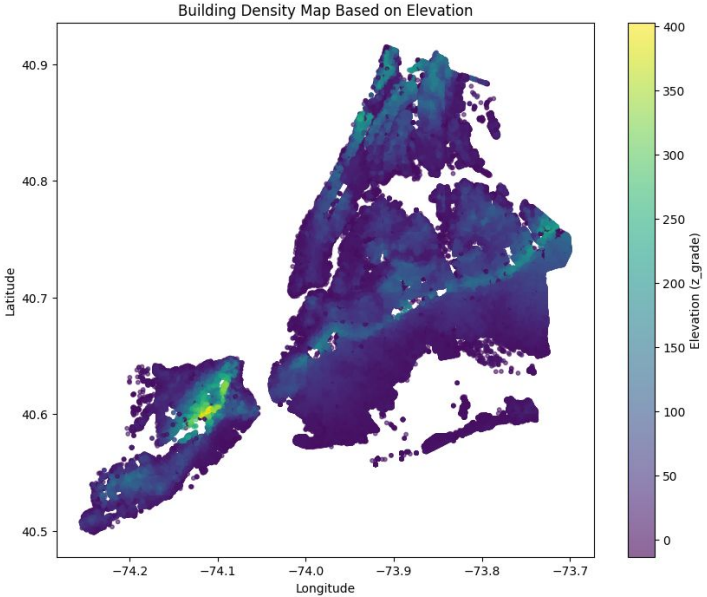


2022

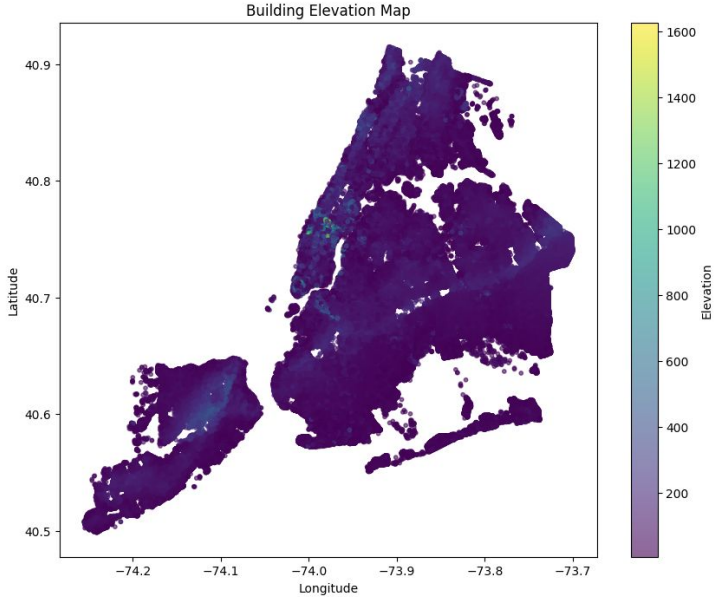


Data Available: Elevation/NYC Data

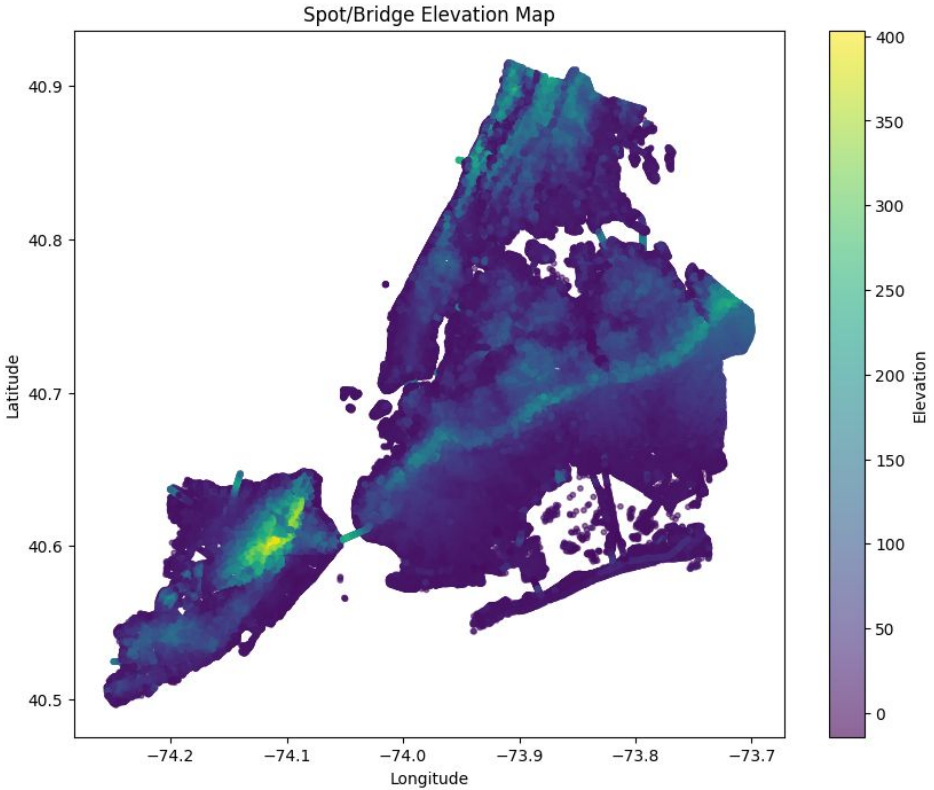
Building Data Collected

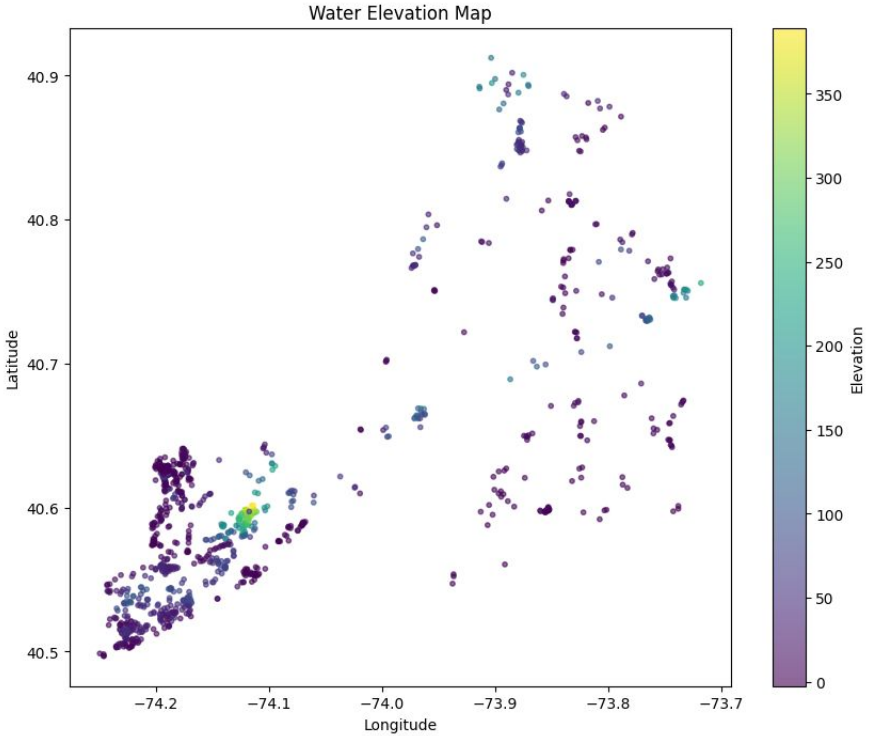


Planimetric Data



Other Elevation





Model Building

The bad models:

Linear Regression between different bands saw no more than a 0.05 R^2

Multiple Linear Regression had no more than 0.1 R^2 value

We have learnt:

This data has non linear complex patterns that needs more advanced ML models

Model Building (Baseline RF)

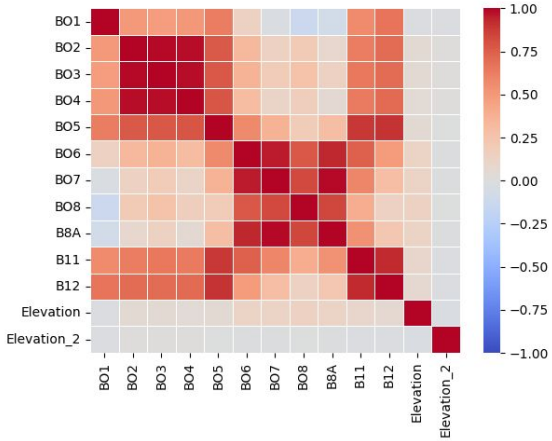
Build a random forest with all columns vs high correlation matrix dropped.

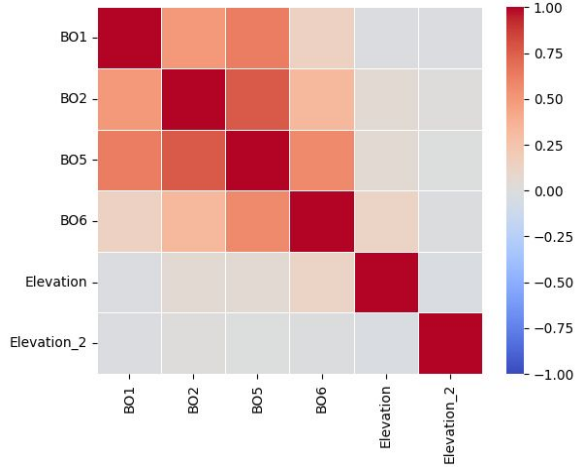
This was a standard RF with default values in the scikit learn RandomForestRegressor() package

- Max_depth: max
- N_estimators: 100
- Max_features: max

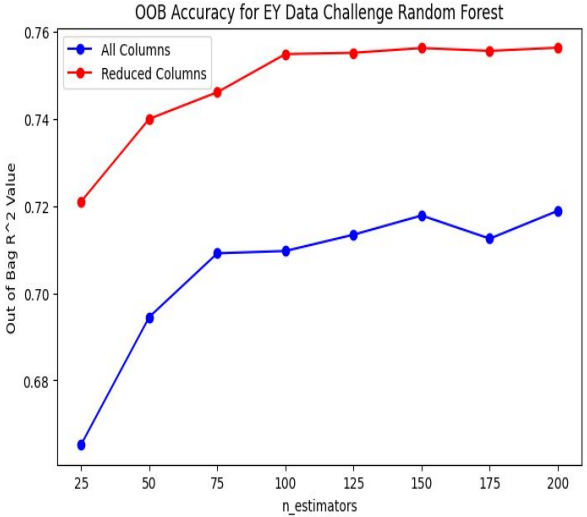
We got a R^2 of 0.71 for all column (train/test)

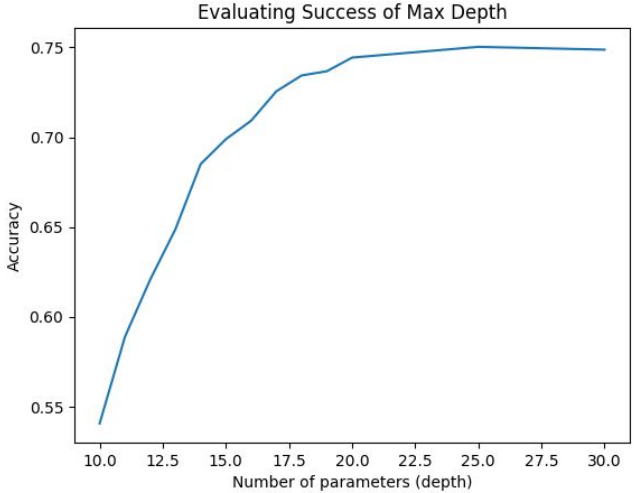
We got a R^2 of 0.75 for correlation dropped column (train/test)

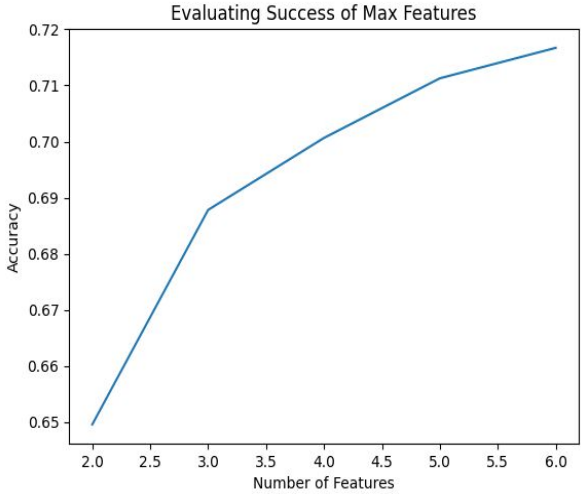




Random Forest Expanded: Hyperparameter Tuning







This is an exhaustive search - we plan on utilizing random grid search or Bayesian Optimization to streamline this process as we build more models

We compared reduced columns vs all columns and found reduced columns to perform far better

Random Forest Official Submissions

Using the elbow points of hyper parameter tuning:

Using bands ['BO1', 'BO2', 'BO5', 'BO6'] we got an R^2 of 0.6238

Using bands ['BO1', 'BO2', 'BO5', 'BO6',’Elevation] we got an R^2 of 0.6539

We are building a thorough jupyter notebook of how to build out a ML model to be able to look back on in the future

What’s next for the Random Forest Model

Data to include

- Perform analysis on different mathematically created band values such as NDVI, NDWI, and NDBI and see which can be used in RF model
- Incorporate Landsat surface temperature as a predictor
- Train data on the proximity UHI values (more general might see better improvements in official submission)

Model improvement

- Hyperparameter tuning: Bayesian Optimization, randomized grid search
- Evaluation: k-fold CV,

Next Ideas for Model Building

- Gradient Boosting: XGBoost, LightGBM, CatBoost
- Convolutional Neural Network
- Clustering?
- Create buffering averages for the the band data/elevation/NYC open data to then train on the proximity data
- Ensemble learning or stacking models: incorporate multiple models into one