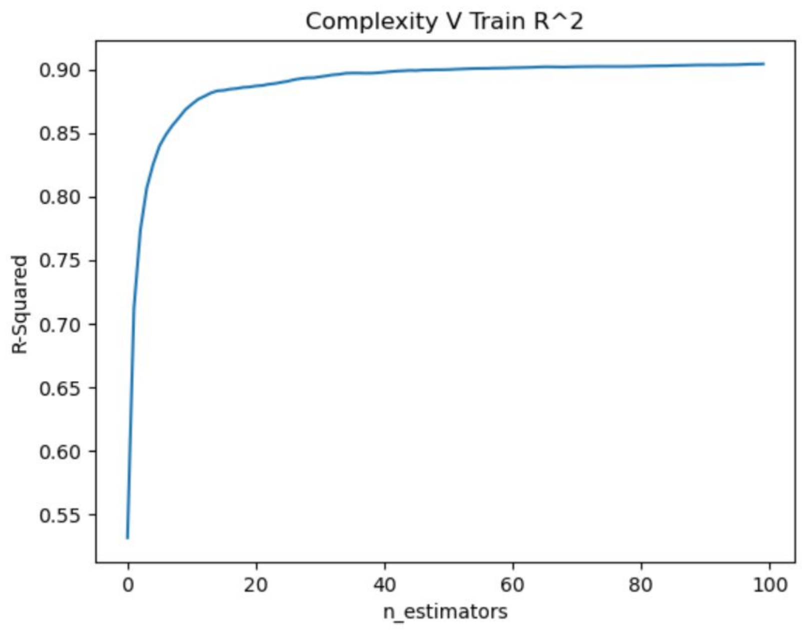
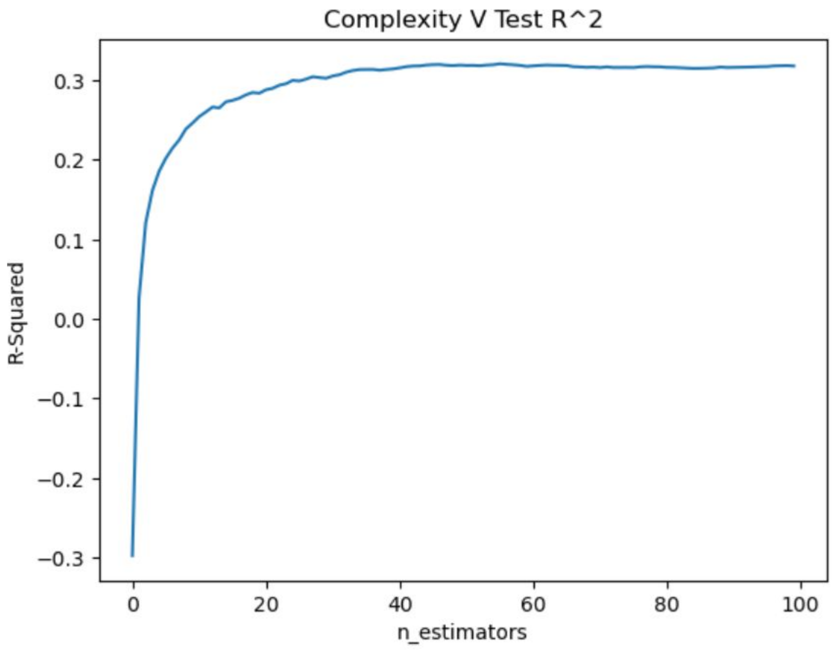
UHI Challenge

Josh, Leon, Bobby

Approach 1- Random Forest Regressor w/ Tiff Data





Approach 2- Neural Network w/ Generated Images

Process-

- Split the data by Manhattan points and Bronx points
- Utilized the contextily, geopandas, and shapely plotting packages to generate birds eye view mappings of the data within the Manhattan bounds
- Trained a neural network (nn_model1) to analyze the image of the Manhattan region
  - Validated with data points from the Bronx
  - Tested with points from Sacramento

```
#get a geoDataFrame for the data and view it
manhattan_gdf = gpd.GeoDataFrame(
    {"geometry": [box(*manhattan_bounds)]},
    crs="EPSG:4326"  # Latitude/Longitude Coordinate System
)


manhattan_gdf = manhattan_gdf.to_crs(epsg=4326)
```

```
nn_model1 = LearnInputsLayers()
```

Approach 2- Manhattan & Training

Training on Manhattan-

- Neural Network taking in RGBA data:
    - Using RGBA values to identify roads, water, vegetation, infrastructure, ect
    - Trains to meet data supplied in 'Training_data_uhi_index_2025-02-18.csv'
- Outputs an estimation of UHI based on RGBA

```
class LearnInputsLayers(nn.Module):
    def __init__(self):
        super(LearnInputsLayers, self).__init__()
        self.hidden_layer = nn.Linear(4, 5) #take
        self.activation = nn.ReLU()
        self.next_layer = nn.Linear(5, 7)
        self.output_layer = nn.Linear(7, 1) #take
        #no activation after output layer

    def forward(self, x):
        x = self.hidden_layer(x)
        x = self.activation(x)
        x = self.next_layer(x)
        x = self.activation(x)
        x = self.output_layer(x)
        # no activation after output layer
        return x
```

Demonstration

Pixel at (1023, 678) has RGBA of [0.3, 0.1, 0.8, 1.0]. Makes an estimation of UHI: 1.05099...

Apply learning model

Improve estimation through training data

Approach 2- Cont.

Validated the data with points front the Bronx

- Used eval to determine loss through MSE
    - Relatively low
- Now the big test was using the data on a west coast city

```python
with torch.no_grad():  # Disable gradient computation for testing
    nn_model1.eval()  # Set the model to evaluation mode
    val_preds = nn_model1(dataB)  # Get predictions on the validation set
    val_loss = criterion(val_preds, targetsB)  # Calculate loss
    print(f'Validation Loss: {val_loss.item():.4f}')
```

✓ 0.0s

```
Validation Loss: 0.0003
```

Tale of the Tape: NY vs Sacramento

vs.

Data supplied by https://urbanheat-smaqmd.hub.arcgis.com/datasets/e1e96c03b4ae44139edc87abfd487a3a/about.

Sacramento Data

Required some levels of cleaning:

- CSV from the The Capital Region Urban Heat Island Mitigation Project
  - https://urbanheat-smaqmd.hub.arcgis.com/datasets/e1e96c03b4ae44139edc87abfd487a3a/about.
- Dropping irrelevant columns
  - Data provided was in degrees celsius
  - UHI calculated by temp/(mean-temp) to scale
- Image from Aerial Archives:
  - https://www.aerialarchives.com/Aerial-Maps-of-Sacramento.htm.

Approach 2 Tested- Sacramento

Despite the Numerous differences between the data values and images, the model still performed relatively well:

```
#try eval on model
with torch.no_grad():  # Disable gradient computation for testing
    nn_model1.eval()  # Set the model to evaluation mode
    val_preds = nn_model1(dataS)  # Get predictions on the validation set
    val_loss = criterion(val_preds, targetsS)  # Calculate loss
    print(f'Validation Loss: {val_loss.item():.4f}')
```

✓ 0.0s

```
Validation Loss: 0.0087
```

Approach 3: Fully Connected Neural Network

- Used training and validation csvs
- Preprocessing: extracted hour, month, day of week
  - Removed unnecessary columns
- Created own time-based values
  - Hour : 0-23 / Month: 1-12/ day of week: 0-6
- Architecture:
  - 3 hidden layers
  - 1 output layer (linear activation for regression)
  - Adam optimizer
  - 100 epochs with 20% validation split

Continued

- Model predicts UHI values based on time patterns
- Predictions were stored on a csv
- Error metrics:

```
### Model Evaluation on Training Data ###
Mean Absolute Error (MAE): 0.1466
Mean Squared Error (MSE): 0.0328
Root Mean Squared Error (RMSE): 0.1812
R² Score: -0.0069
```

UHI Index	hour	month	day_of_week
1.0460359	6	5	1
1.0460359	6	5	1
1.0460359	6	5	1
1.0460359	6	5	1
1.0460359	6	5	1
1.0460359	6	5	1
1.0460359	6	5	1
1.0459907	7	5	1
1.0459907	7	5	1
1.0459907	7	5	1
1.0459907	7	5	1
1.0459907	7	5	1
1.0459907	7	5	1
1.0459907	7	5	1
1.0459907	7	5	1
1.0459907	7	5	1

What this shows us

Infrastructure matters!

- Though this model just considers colors, the UHI is affected by proximity to certain colors:
  - These colors represent certain types of locations such as buildings, waterways, fields, ect
  - Location types are an effective predictor of UHI