Predicting Population

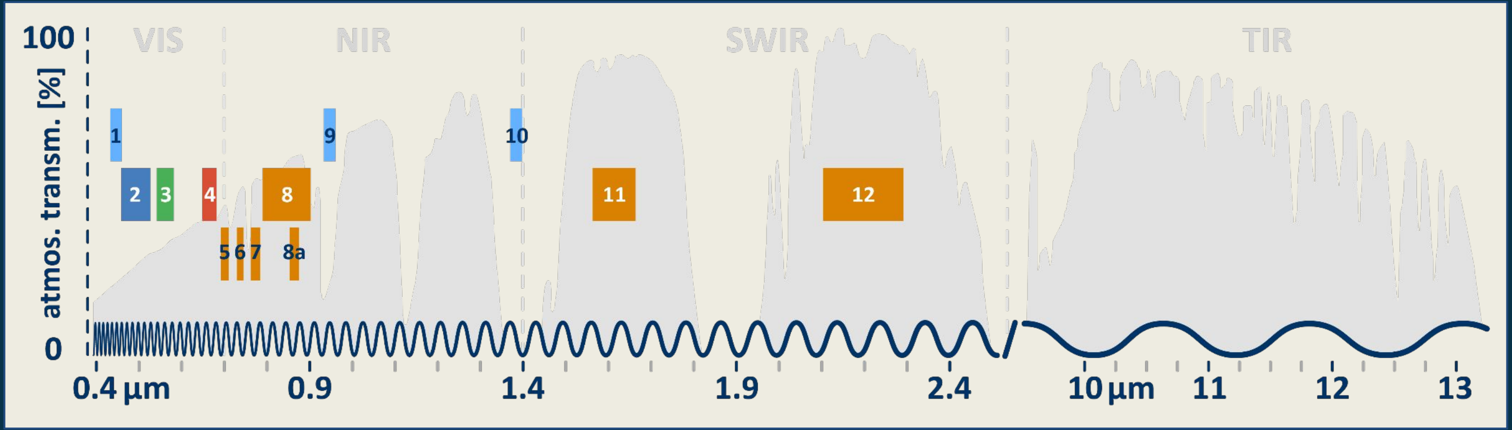
Emma Fiedler, Franziska Funk,
Alexander Gerken, Jonah Lux, Felice de Pinto

01 Motivation

https://www.eea.europa.eu/en/analysis/maps-and-charts/modelled-number-of-people-flooded-across-europes-coastal-areas-in-1961-1990-and-in-the-2080s#references-and-footnotes

02 The Data

02 The Data: Satellites



https://blogs.fu-berlin.de/reseda/files/2018/05/sentinel_2_channels.png

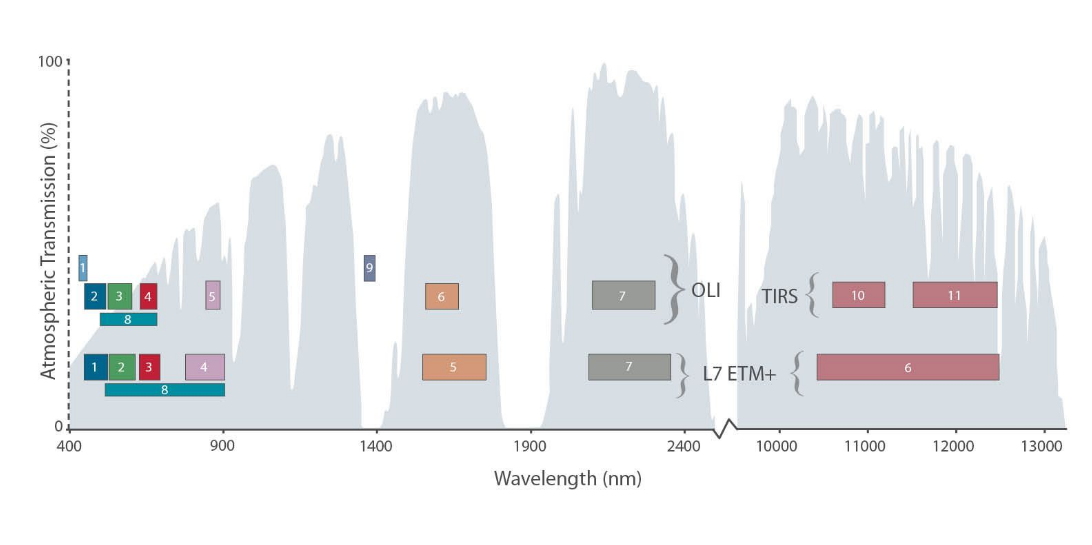
Sentinel 2 -

11 bands (excluding 9 + 10)

02 The Data: Satellites

Landsat 8 -

1 thermal infrared band

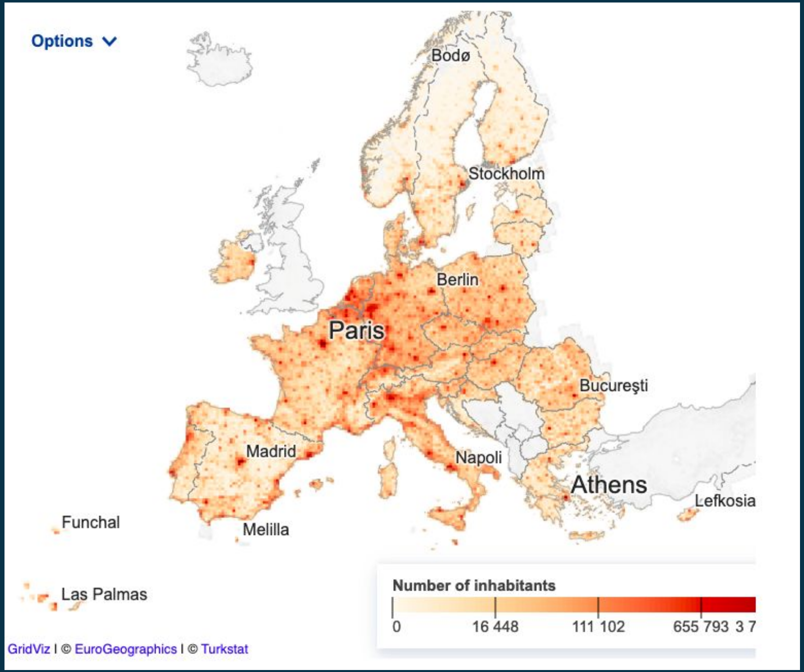


https://landsat.gsfc.nasa.gov/wp-content/uploads/2021/12/ETMvOLI-TIRS-web_Feb20131.jpg

02 The Data: Population

Population data from EUROSTAT

- 1km resolution
- converted to 5kmx5km tiles



https://ec.europa.eu/eurostat/web/gisco/geodata/grids

02 The Data

Total: 1000 tiles (5km x 5km), 20m x 20m pixels - **2GB**

log-transformed

+ 9 bands

03 Our Approaches

03 Our Approaches

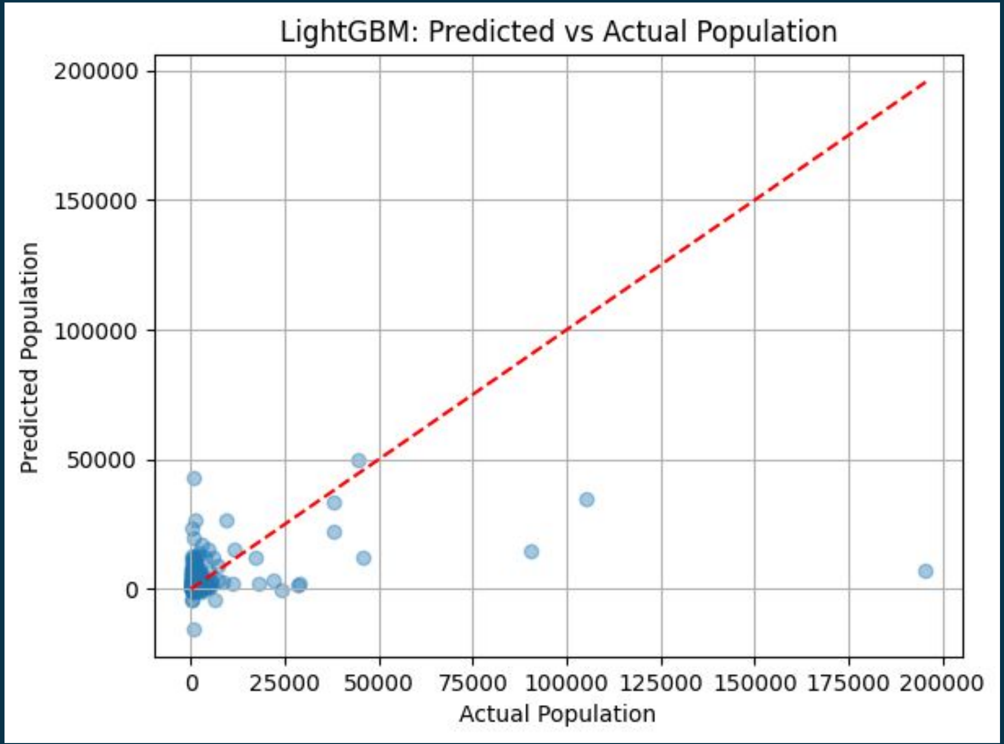
Different approaches:

1. Simple Regression
2. Clustering + Regression
3. CNN with ResNet

➢ on all of Europe (+1000 tiles)

03 Approach 1: Simple Regression

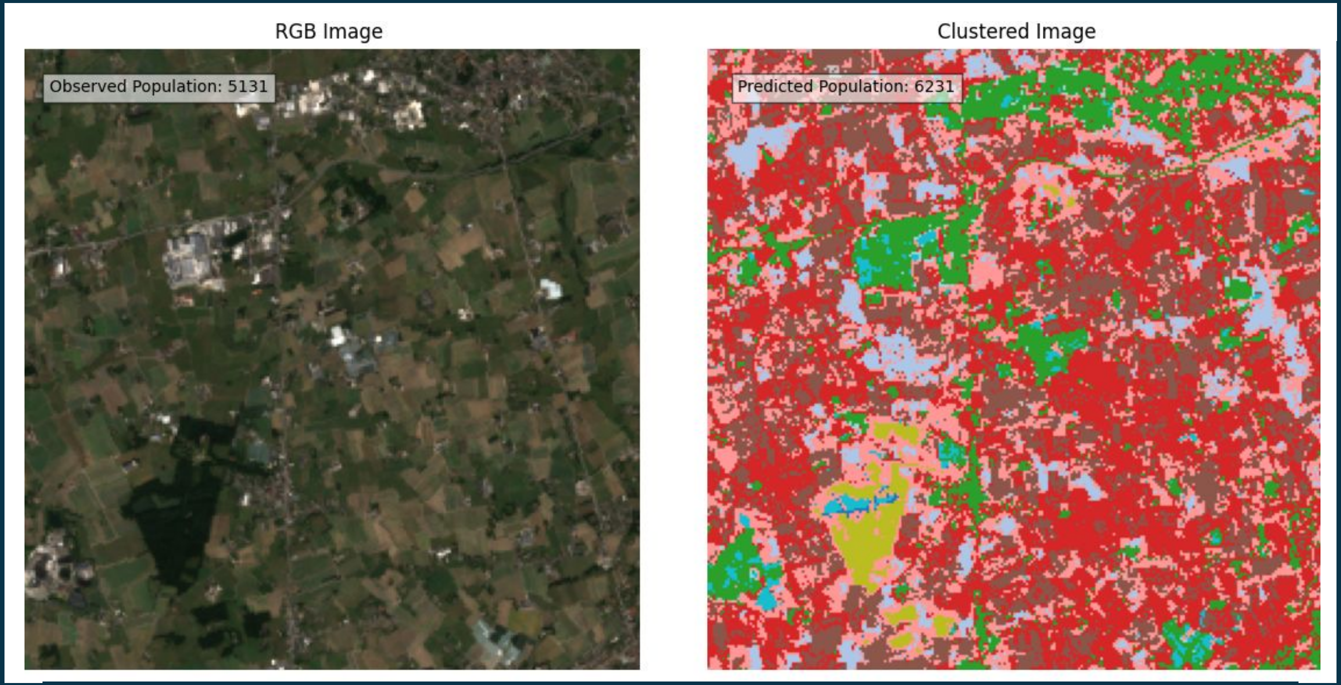
03 Simple Regression



- Averaging over band values in each tile
- LGBMRegressor
- R^2 score: 0.1

03 Approach 2: Regression + Clustering

03 Regression + Clustering

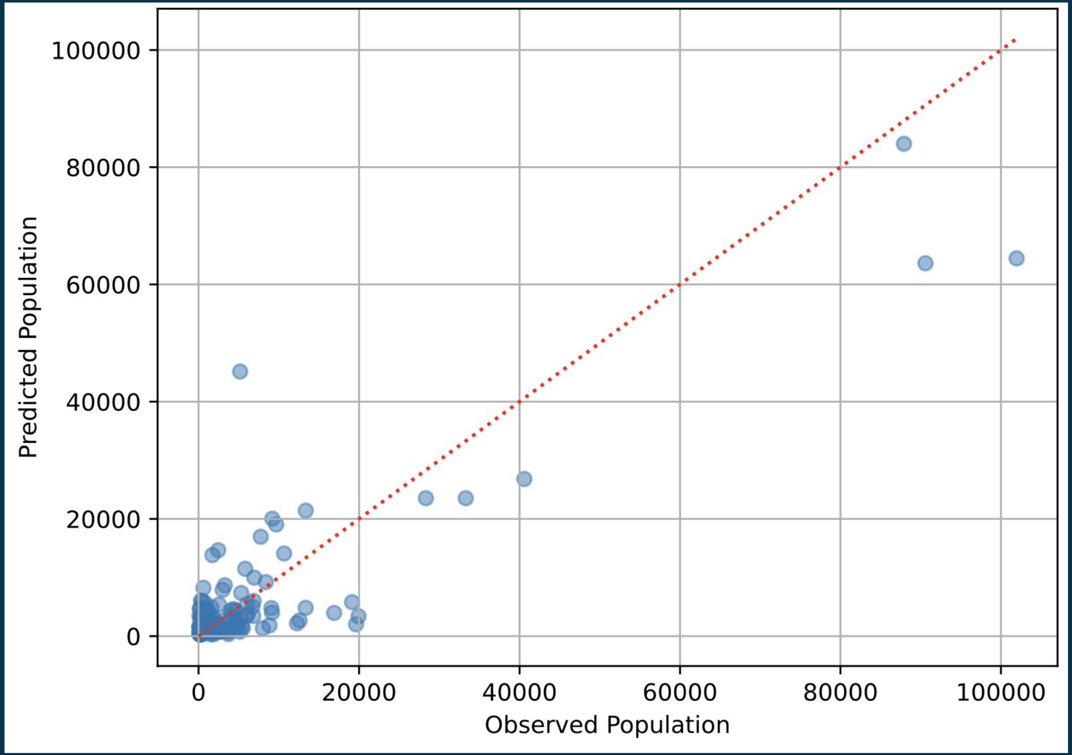


03 Regression + Clustering

Different models:

- **Clusterers:** GMM, K-Means, Spectral Clusterer, Fuzzy Clustering
- Optimization of **number of clusters:** 5-100
- **Regressors:** Linear Regressor, lgbm, XGBoost, TensorFlow, MLP Regressor
- **Hyperparameter** Optimization: Bayes & RandomSearch
- all of **Europe**, only **coast** of Italy

03 Regression + Clustering



R2 score: 0.52

03 Approach 3: CNN

03 CNN - ResNet

Open Access | Feature Paper | Article

Mapping Multi-Temporal Population Distribution in China from 1985 to 2010 Using Landsat Images via Deep Learning

by Haoming Zhuang [1], Xiaoping Liu [1,2,*], Yuchao Yan [3], Jinpei Ou [1], Jialyu He [1] and Changjiang Wu [1]

[1] Guangdong Key Laboratory for Urbanization and Geo-Simulation, School of Geography and Planning, Sun Yat-sen University, Guangzhou 510275, China

[2] Southern Marine Science and Engineering Guangdong Laboratory (Zhuhai), Zhuhai 519082, China

[3] Sino-French Institute for Earth System Science, College of Urban and Environmental Sciences, Peking University, Beijing 100091, China

[*] Author to whom correspondence should be addressed.

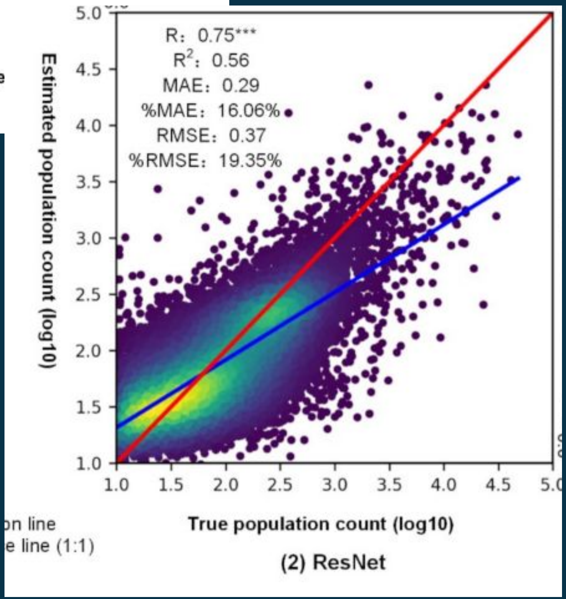
Remote Sens. **2021**, *13*(17), 3533; https://doi.org/10.3390/rs13173533

Submission received: 3 August 2021 / Revised: 2 September 2021 / Accepted: 3 Septembe

Published: 6 September 2021

What do we do?

- ➢ Regressing population with CNN ResNet-50
- ➢ 6 Spectral Bands from Sentinel 2
- ➢ Loss function: Log-Cosh Loss
- ➢ SDG with a momentum of 0.9 and a learning rate of 0.0001



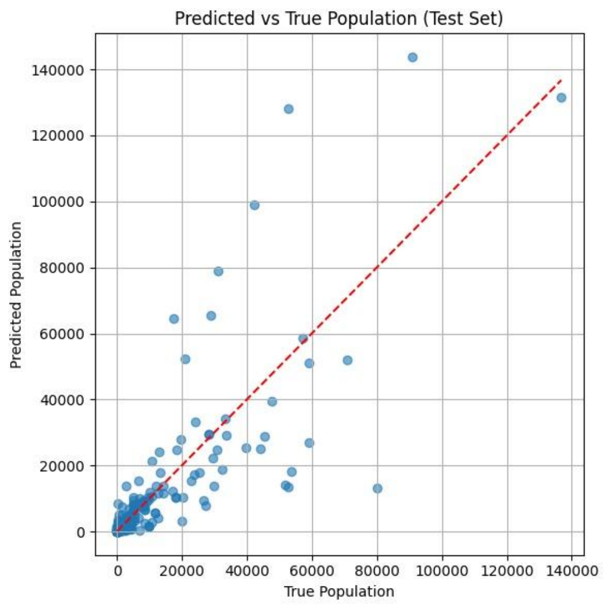
03 CNN - ResNet

We optimized our work:

- ➢ logging the population true data for better data process
- ➢ resizing tiles to 224 x 224 pixel and normalizing bands
- ➢ data augmentation (horizontal and vertical flipping)
- ➢ 2400 tiles all around europe on google colab GPU (3 hours)
- ➢ training (70%), validation (10%), test (20%)
- ➢ 25 epoch with 5 patience and batch size 32

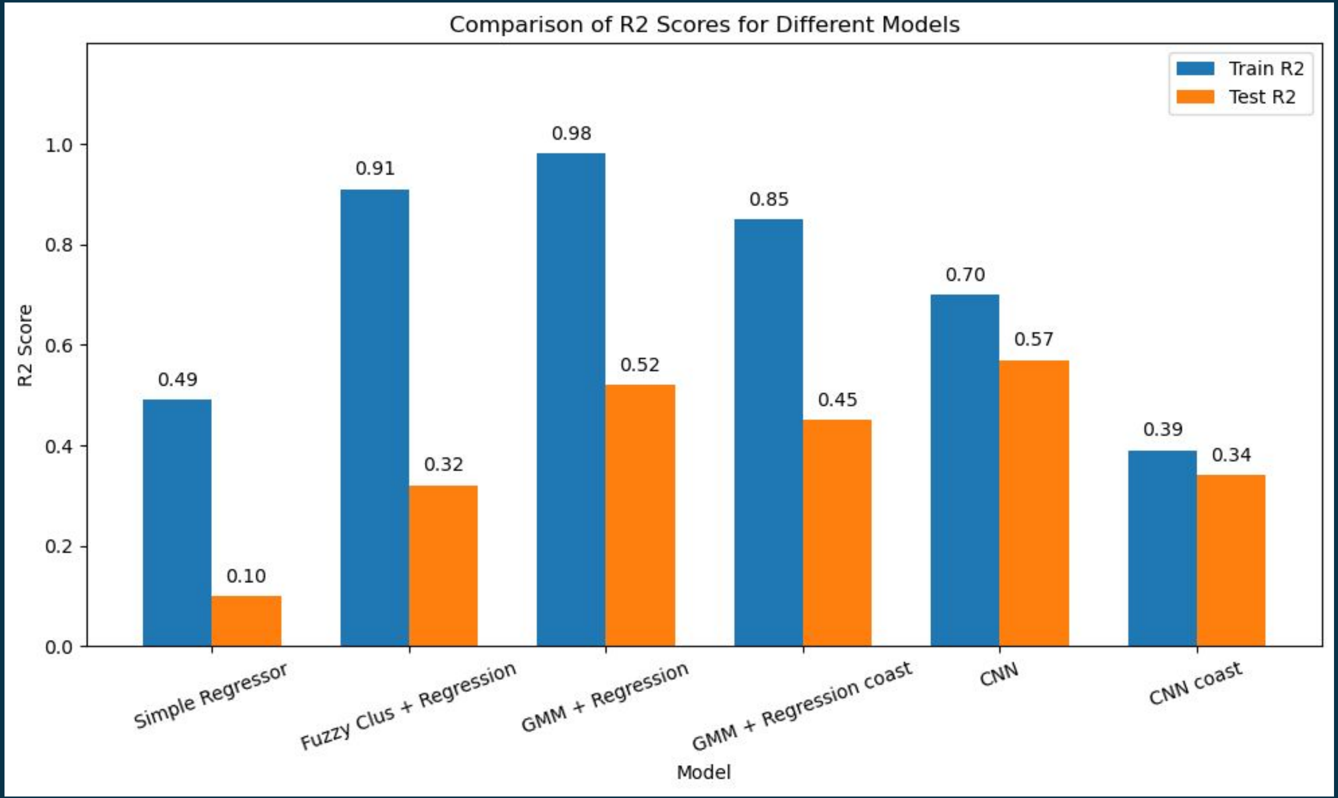
Our results on test data:

- ➢ $R^2 = 0.57$
- ➢ MARE = 70%



04 Comparison

04 Comparison



- best performance: CNN
- but could be better…
- dependent on random state

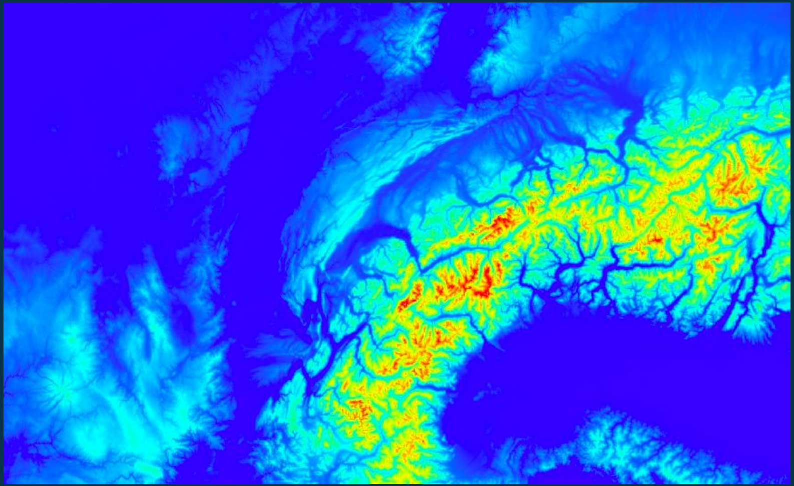
05 Adding more satellite data

05 Adding more satellite data

Night Light

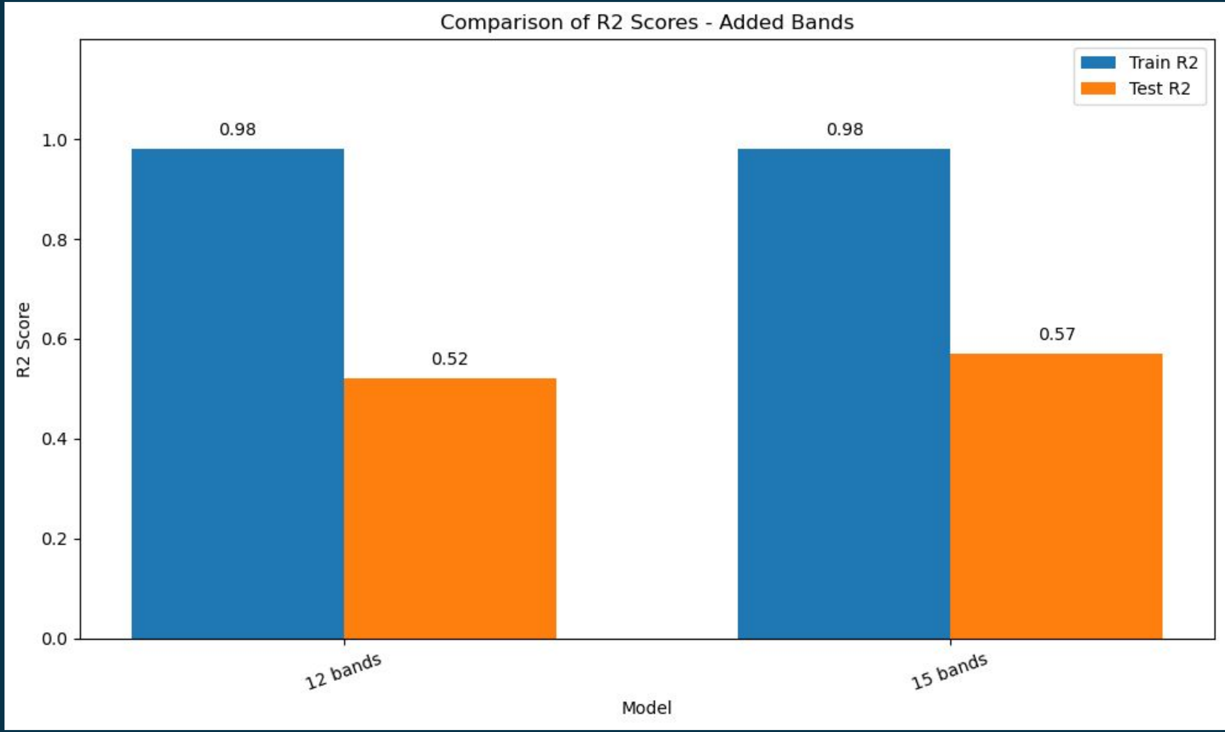
https://developers.google.com/earth-engine/datasets/catalog/NOAA_VIIRS_001_VNP46A2?hl=de

Elevation and Slope



https://www.eorc.jaxa.jp/ALOS/en/dataset/aw3d30/aw3d30_e.htm

05 Adding more satellite data



- better but still not good..
- dependant on random state

06 Conclusion and Outlook

06 Conclusion

- ➢ 3 different approaches
- ➢ Overall no reliable population prediction possible
- ➢ Pre-trained CNN performs the best
- ➢ Several struggles:
  - R^2 dependent on random seed
  - Extremely long computational time
  - …

What could improve the results?

- More tiles + tile neighborhood
- More computer power (faster GPU)
- Different information, e.g.:
  - ➢ mobile data
  - ➢ pollution
  - ➢ data from different seasons

Thanks for your Attention
And Special Thanks to Aslak!

APPENDIX

What the data looks like

x	y	pop	F0	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
5262500.0	3912500.0	113.0	0.0	0.0384	0.002	0.0	0.0116	0.0004	0.0	0.0036	0.136	0.1952	0.0772	0.0604	0.0336
4892500.0	1752500.0	3626.0	0.0	0.0056	0.136	0.0	0.0872	0.1476	0.0	0.1308	0.0004	0.0272	0.0	0.4112	0.0
3422500.0	1872500.0	58546.0	0.0	0.0	0.0	0.0	0.0	0.0776	0.0	0.1968	0.0	0.0	0.0	0.008	0.0
4572500.0	4232500.0	3821.0	0.0	0.0128	0.114	0.0	0.004	0.0	0.0	0.0168	0.01	0.234	0.0544	0.0064	0.1308
2867500.0	2267500.0	336.0	0.0	0.0024	0.2852	0.0	0.0548	0.0252	0.0	0.0212	0.0012	0.0964	0.0012	0.2124	0.0012
3157500.0	3377500.0	1172.0	0.0	0.038	0.024	0.0	0.0032	0.0	0.0	0.04	0.0876	0.2784	0.0084	0.0516	0.022
5447500.0	1612500.0	3855.0	0.184	0.0004	0.1924	0.0	0.0008	0.0428	0.0264	0.2024	0.0016	0.0	0.0	0.062	0.0
4412500.0	3907500.0	552.0	0.2676	0.0032	0.0768	0.0	0.0008	0.0	0.0008	0.0756	0.0136	0.0908	0.0292	0.0064	0.0696
3522500.0	2522500.0	809.0	0.0	0.0	0.0152	0.0	0.0744	0.0004	0.0	0.002	0.0032	0.09	0.346	0.0896	0.0112
3372500.0	2297500.0	2910.0	0.0	0.0064	0.0828	0.0	0.3808	0.0012	0.0	0.0256	0.0	0.1532	0.016	0.1492	0.0188
4407500.0	3167500.0	984.0	0.0	0.0024	0.0196	0.0	0.0344	0.008	0.0	0.0088	0.1996	0.1012	0.28	0.094	0.0884
4202500.0	2697500.0	42983.0	0.0	0.0908	0.0152	0.0	0.0104	0.0048	0.0	0.0428	0.138	0.1708	0.0804	0.0912	0.0252
4197500.0	3092500.0	466.0	0.0	0.0	0.1244	0.0	0.0108	0.0	0.0	0.0188	0.0016	0.0628	0.0308	0.0208	0.132
5277500.0	4392500.0	1073.0	0.7376	0.0	0.0096	0.0	0.0	0.0	0.0008	0.0024	0.0	0.0024	0.0224	0.0	0.0596
4272500.0	3512500.0	3408.0	0.0	0.4124	0.0	0.0	0.0012	0.0	0.0	0.0096	0.0492	0.288	0.0056	0.0148	0.0092
3527500.0	2662500.0	669.0	0.0	0.0792	0.0032	0.0	0.0028	0.0212	0.0	0.0316	0.3128	0.0852	0.0076	0.1516	0.0056
3047500.0	3542500.0	165.0	0.0	0.0	0.8052	0.0	0.0012	0.0	0.0	0.0	0.0	0.0104	0.0008	0.0	0.1232
5357500.0	3767500.0	546.0	0.0	0.0368	0.006	0.0	0.0388	0.0056	0.0	0.0088	0.0752	0.1412	0.0048	0.07	0.0036
4547500.0	2537500.0	1309.0	0.0	0.0516	0.002	0.0	0.0232	0.016	0.0	0.1276	0.0144	0.0964	0.0	0.0752	0.0
5512500.0	1727500.0	324.0	0.1576	0.0	0.0	0.0	0.0	0.5116	0.066	0.0236	0.0	0.0	0.0	0.0092	0.0
4297500.0	2792500.0	4289.0	0.0	0.3092	0.014	0.0	0.016	0.0	0.0	0.014	0.2128	0.0736	0.0252	0.0496	0.038
4932500.0	2522500.0	302.0	0.0	0.004	0.0036	0.0	0.3824	0.0	0.0	0.0012	0.0	0.058	0.4016	0.0064	0.0668
3277500.0	1707500.0	196.0	0.0	0.0	0.0	0.0	0.0	0.702	0.0	0.11	0.0	0.0	0.0	0.0084	0.0

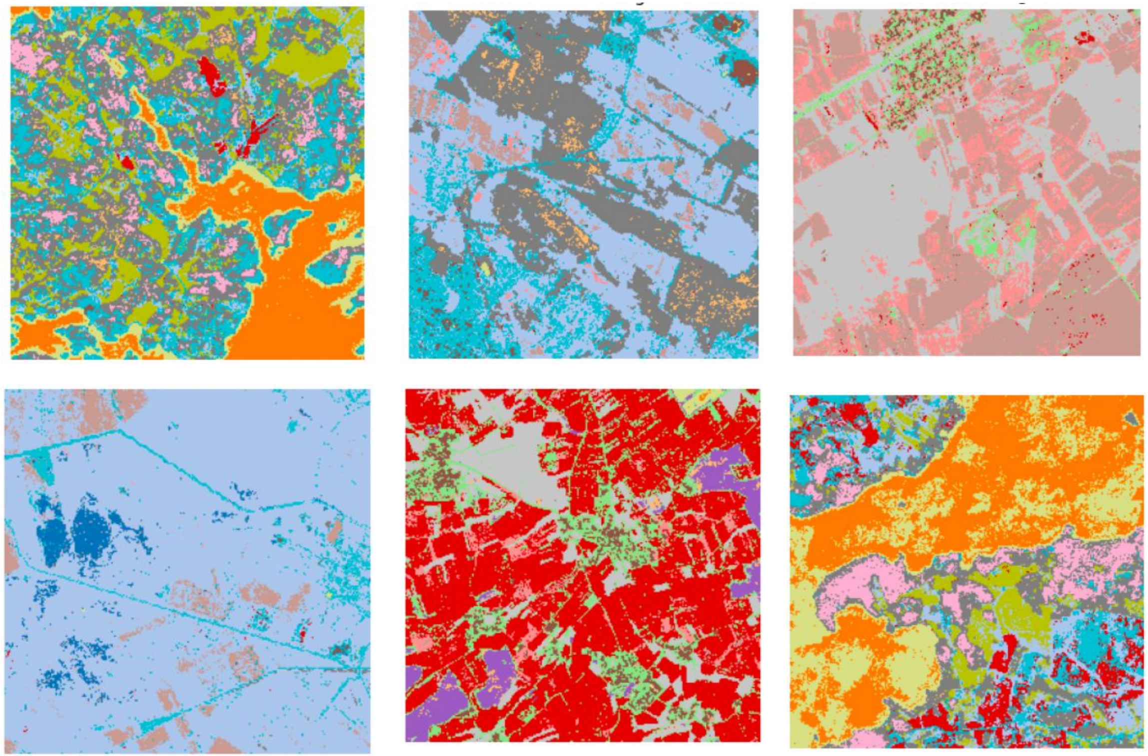
.

.

.

Cluster percentages for each tile

Examples of clustered tiles

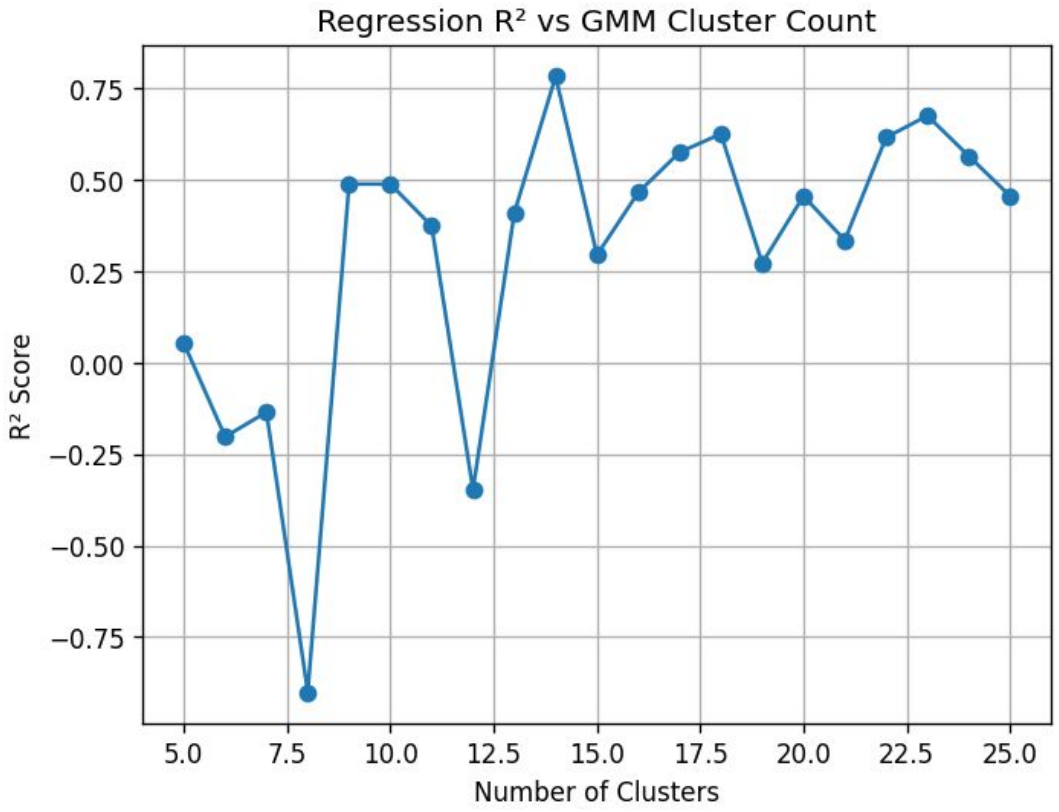


Clustering + Regression Recipe

1. Log-transform satellite data and train Clusterer for n clusters
2. Cluster all tiles with trained Clusterer and calculate cluster percentages
3. Match population data to each tile
4. Split data (population + cluster percentages) into test and train
5. Train Regressor on train data, using hyperparameter optimization
6. Predict population for test data with trained Regressor
7. Loop over steps 1. to 6. to find best number of clusters

(not reliable, changes with random state of split)	Bands included	Test R^2	Train/Validate R^2	Number Clusters
Simple Regressor	12	0.10	0.49	-
Fuzzy Clustering + lgbm regressor	12	0.32	0.91	18
GMM + LinearRegressor	12	0.39	0.43	12
GMM + lgbm regressor	12	0.29	0.92	18
	12 (coast)	0.48	0.98	18
	15	0.46	0.82	14
GMM + XGboost	12	0.52	0.98	90
	12 (coast)	0.45	0.85	65
	15	0.57	0.98	17
CNN ResNet	6	0.57	0.70	-
	6 (coast)	0.34	0.39	-
	15	0.38	0.43	-

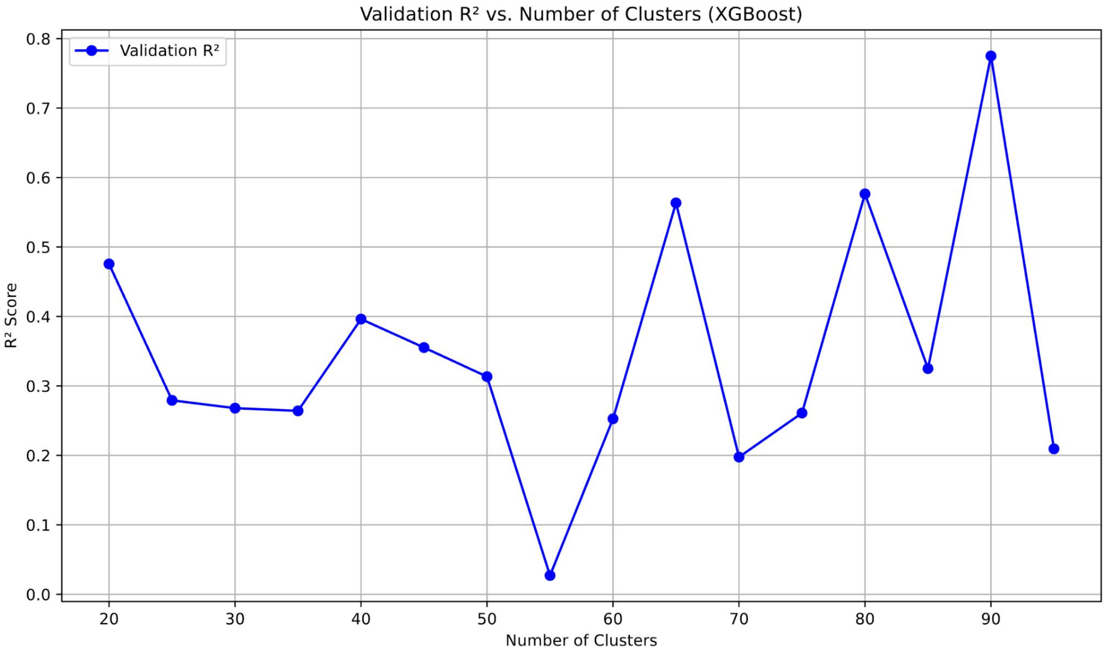
Finding the right cluster amount



example for lgbm Regressor with 15 bands

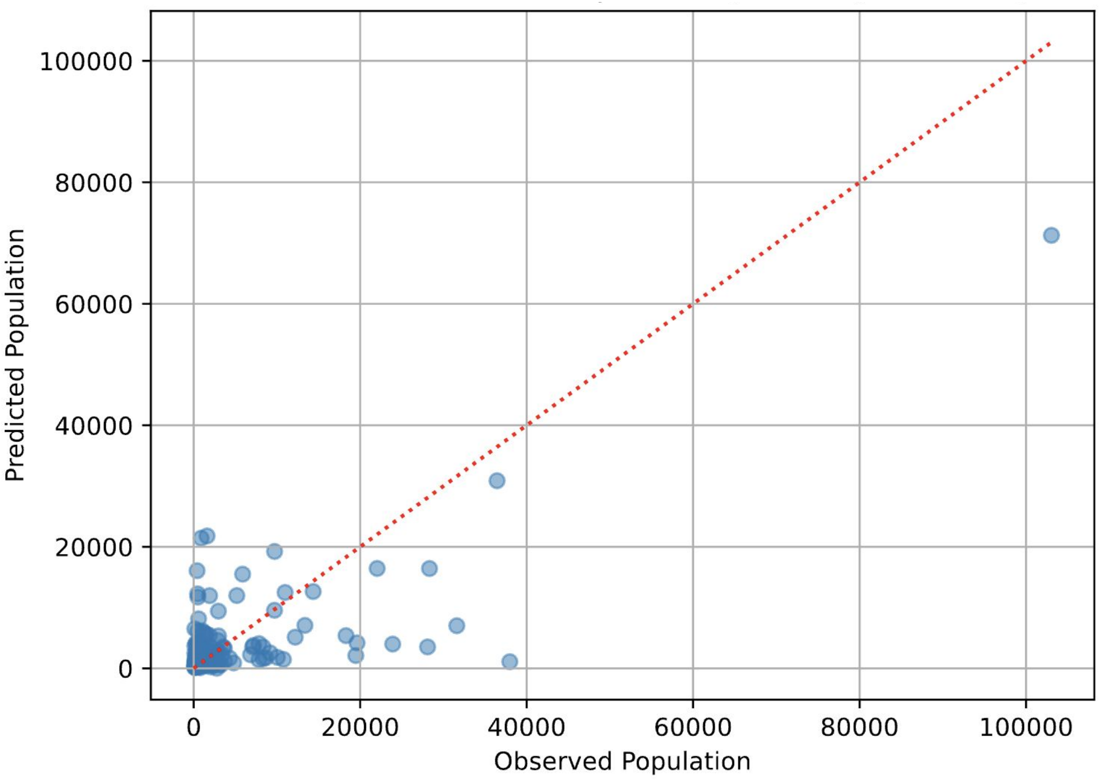
Note that random state of splitting data into test and train changes this

Finding the right cluster amount



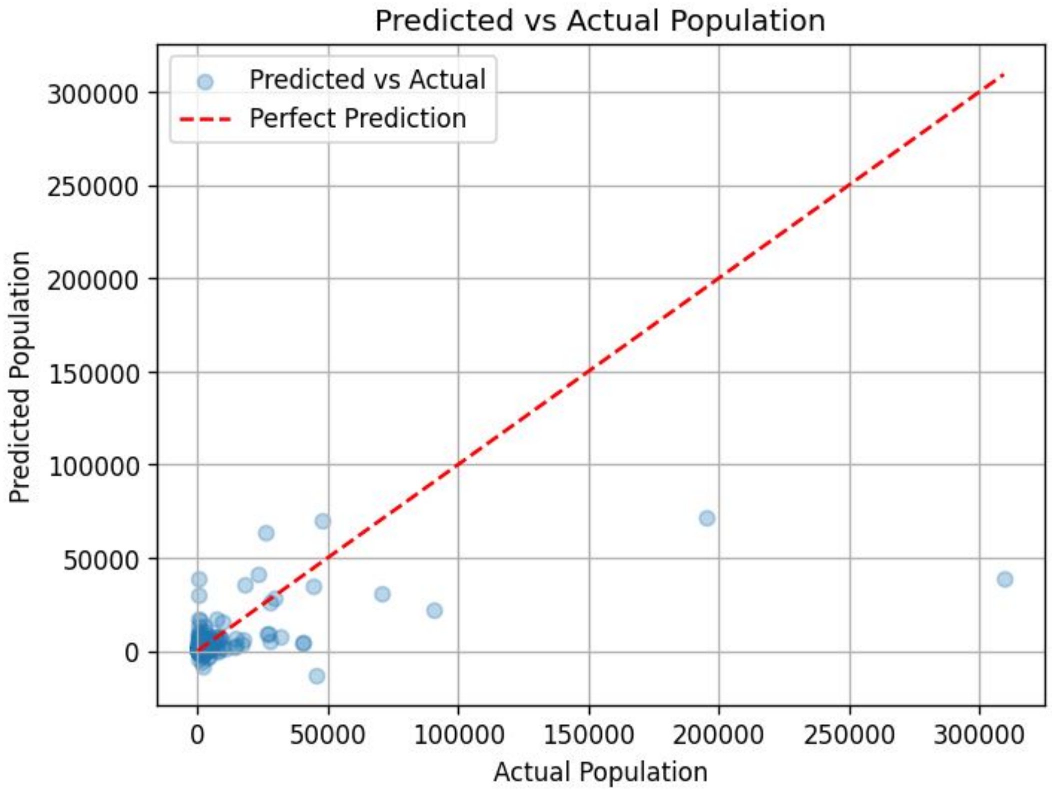
example for XGBoost Regressor with 12 bands

Predicted vs. observed population



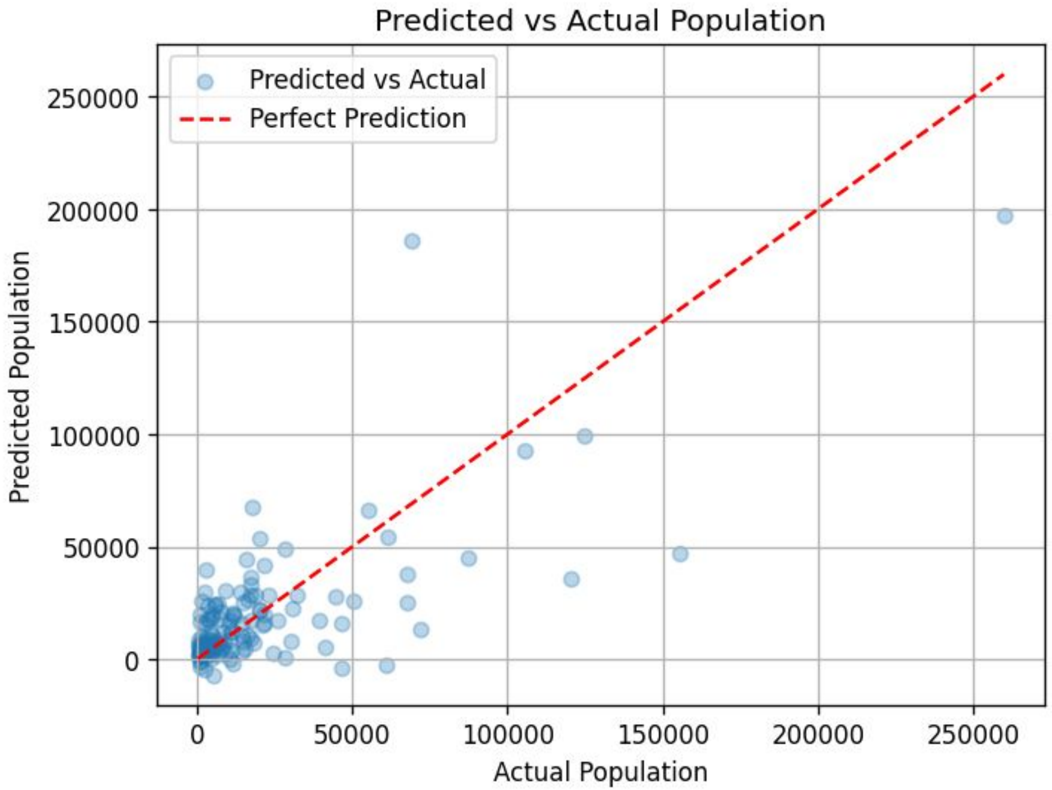
example for XGBoost Regressor
with 15 bands
$R^2 = 0.56$

Predicted vs. observed population



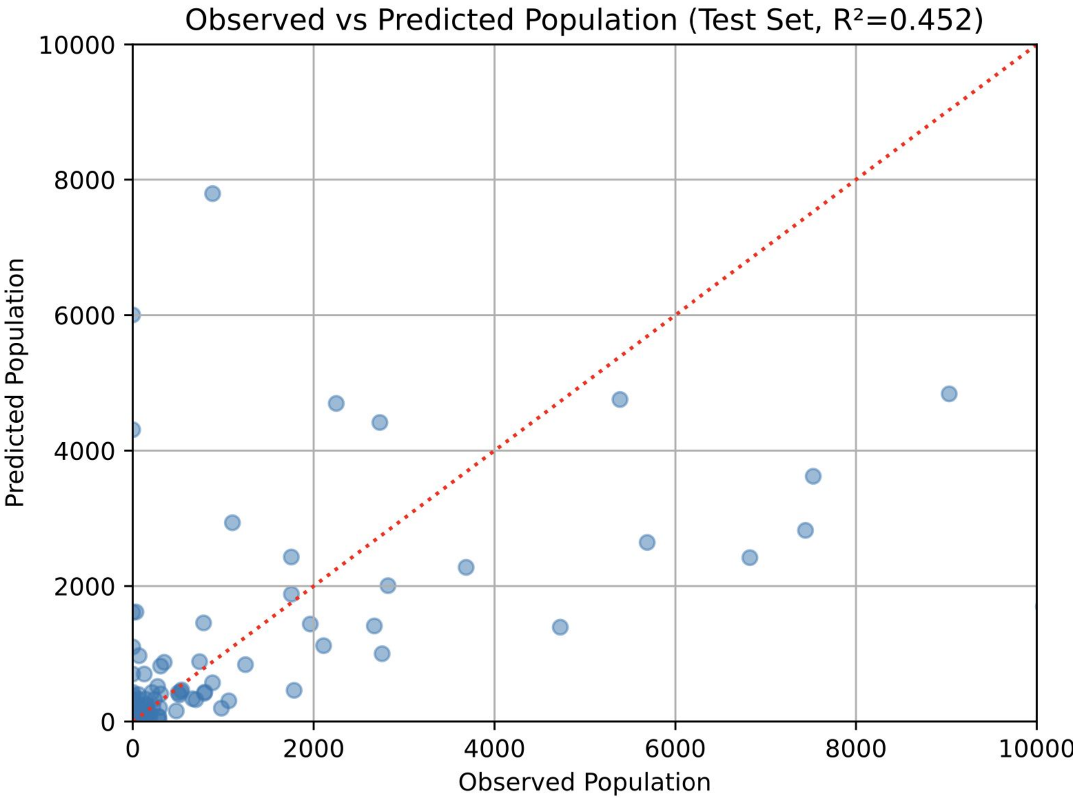
example for lgbm Regressor with 12 bands
$R^2 = 0.29$

Predicted vs. observed population



example for lgbm Regressor with
12 bands coastal data
R^2 = 0.48

Predicted vs. observed population



example for XGBRegressor with 12 bands coastal data
R^2 = 0.45

03 CNN - ResNet - Other trials

- ➢ 5-fold cross-validation (computationally intense and didn't really improve performance >> 5 hours + computer explosion)
- ➢ using 15 bands and 800 tiles (didn't really improve the performance)
- ➢ tile neighborhood (computationally intense and pretty useless with small number of tiles >> 5 hours)
- ➢ 700 coastal tiles

Our results using 15 bands:

- ➢ $R^2 = 0.38$
- ➢ MARE = 150%