EAEE E4000 Course Project

Predicting Land Surface Temperature from Landscape Metrics

Haokai Zhao

Instructor: Prof. Pierre Gentine

December 2021

Machine Learning for Environmental Engineering and Sciences
Columbia University

Table of Contents

1. Introduction

The urban heat island (UHI) effect, which manifests as a phenomenon that the temperature in the urban area is higher than in the surrounding areas, can exacerbate extreme heat events, which have resulted in an increase of heat-related mortality in recent years (Kleerekoper et al., 2012). The UHI is often formed as a result of increased anthropogenic heat emissions, reduced evaporative cooling, decreased surface permeability, lower surface albedos and narrow urban canyon morphology (Debbage & Shepherd, 2015).

A large number of research have been carried out to investigate surface UHI or the associated land surface temperature (LST) using landscape metrics. Some of these landscape metrics, which can be used to describe landcover composition and landcover configuration, have been highlighted by researchers as important factors influencing the UHI (Debbage & Shepherd, 2015; Galletti et al., 2019; Lu et al., 2020; Shaker et al., 2019; Zhou et al., 2011).

In this project, land surface temperature and landscapes metrics data for Boston area were collected. To develop models for predicting land surface temperature from landscape metrics, two types of machine learning techniques were used: support vector machine (SVM) and XGBoost. The best model achieved a test R^2 of 0.87; tree canopy fraction was identified as the most influential factor.

2. Data and Methods

2.1 Description of Data

A composite dataset (Trlica, 2017), which consists of land surface temperature and several landscape metrics, was collected for the Boston area. The variables used in this study are:

- Albedo (alb), which measures surface reflectivity, with darker surfaces reporting lower albedo scores and brighter surfaces reporting higher albedo scores;
- Tree canopy fraction (can), which measures the percent of tree canopy coverage;
- Impervious surface fraction (isa), which measures the percent of impervious surface coverage;
- Population density (pop), measures the number of persons per squared kilometer;

- Land surface temperature (lst), which measures the land surface temperature in degrees Celsius (°C);
- Land Use/Land Cover (LUfull), which indicates the land use classification.

In the original dataset, all variables were aggregated into 30m raster cells, except for population density (pop), which has a resolution of 1km. To align the variables (layers), they were resampled to the same 30m resolution in the ArcGIS Pro software. In addition, the only categorical variable is Land Use/Land Cover (LUfull), as the others are all numerical variables. Individual maps for each variable are shown in Figure 2-1.

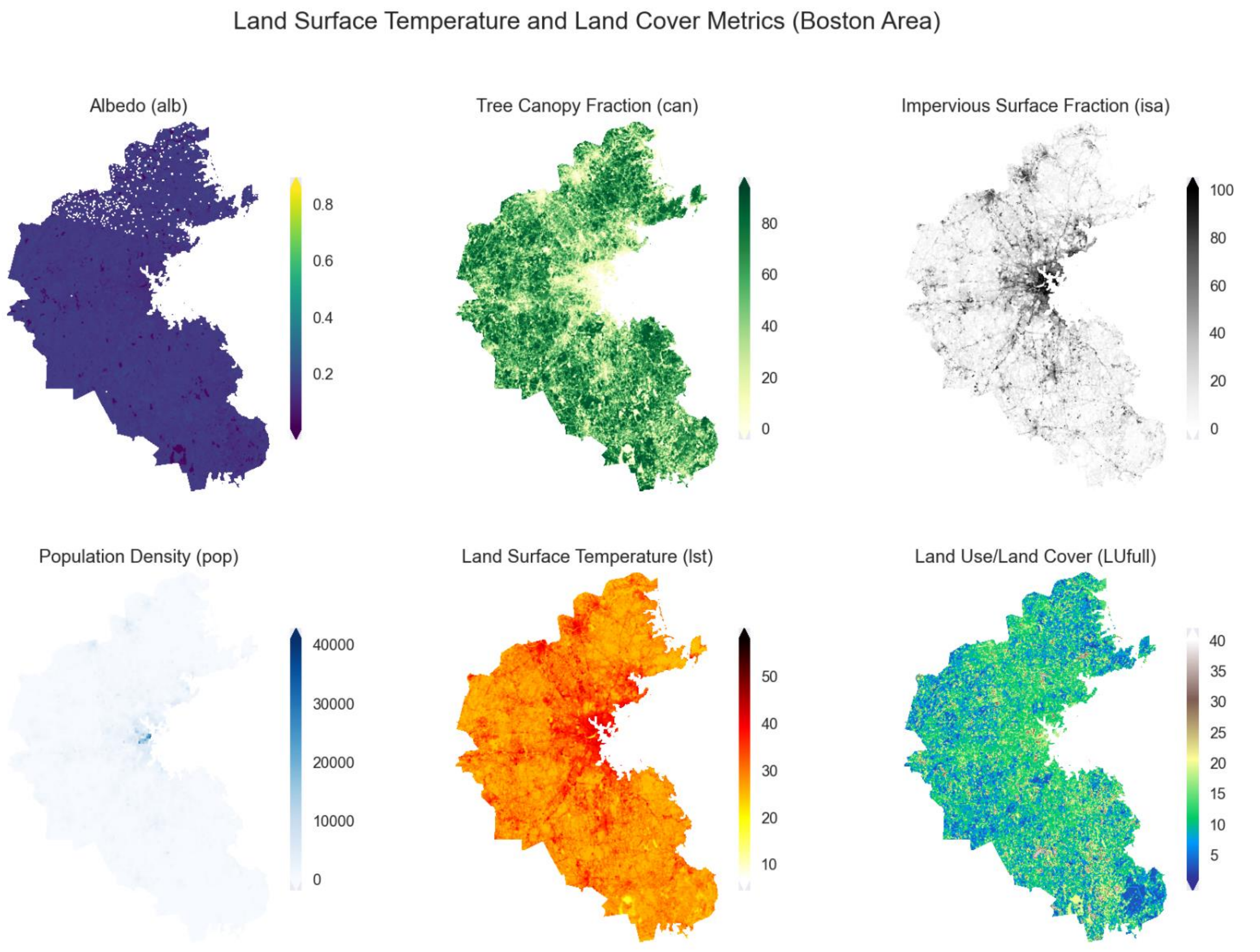

Land Surface Temperature and Land Cover Metrics (Boston Area)
Albedo (alb)
0.8
0.6
0.4
0.2
Tree Canopy Fraction (can)
80
60
40
20
0
Impervious Surface Fraction (isa)
100
80
60
40
20
0
Population Density (pop)
40000
30000
20000
10000
0
Land Surface Temperature (lst)
50
40
30
20
10
Land Use/Land Cover (LUfull)
40
35
30
25
20
15
10
5

Figure 2-1 Land surface temperature and landscape metrics (Boston Areas)

As some machine learning models can not handle categorical variables, here, the LUfull variable was converted to corresponding runoff coefficient values, which were found in literature (NYC DEP, 2012; Nyman et al., 2002). Runoff coefficient is a dimensionless coefficient representing the runoff-to-precipitation ratio. It ranges from 0 to 1 and is higher in areas with low infiltration and high runoff and lower in permeable, well-vegetated areas. Because of the distinct runoff coefficient values associated with various landcovers, this coefficient can be used as an effective indicator of landcover type.

A pairwise scatter plot was created for all variables, as shown in Figure 2-2, and no apparent correlation between variable pairs was found.

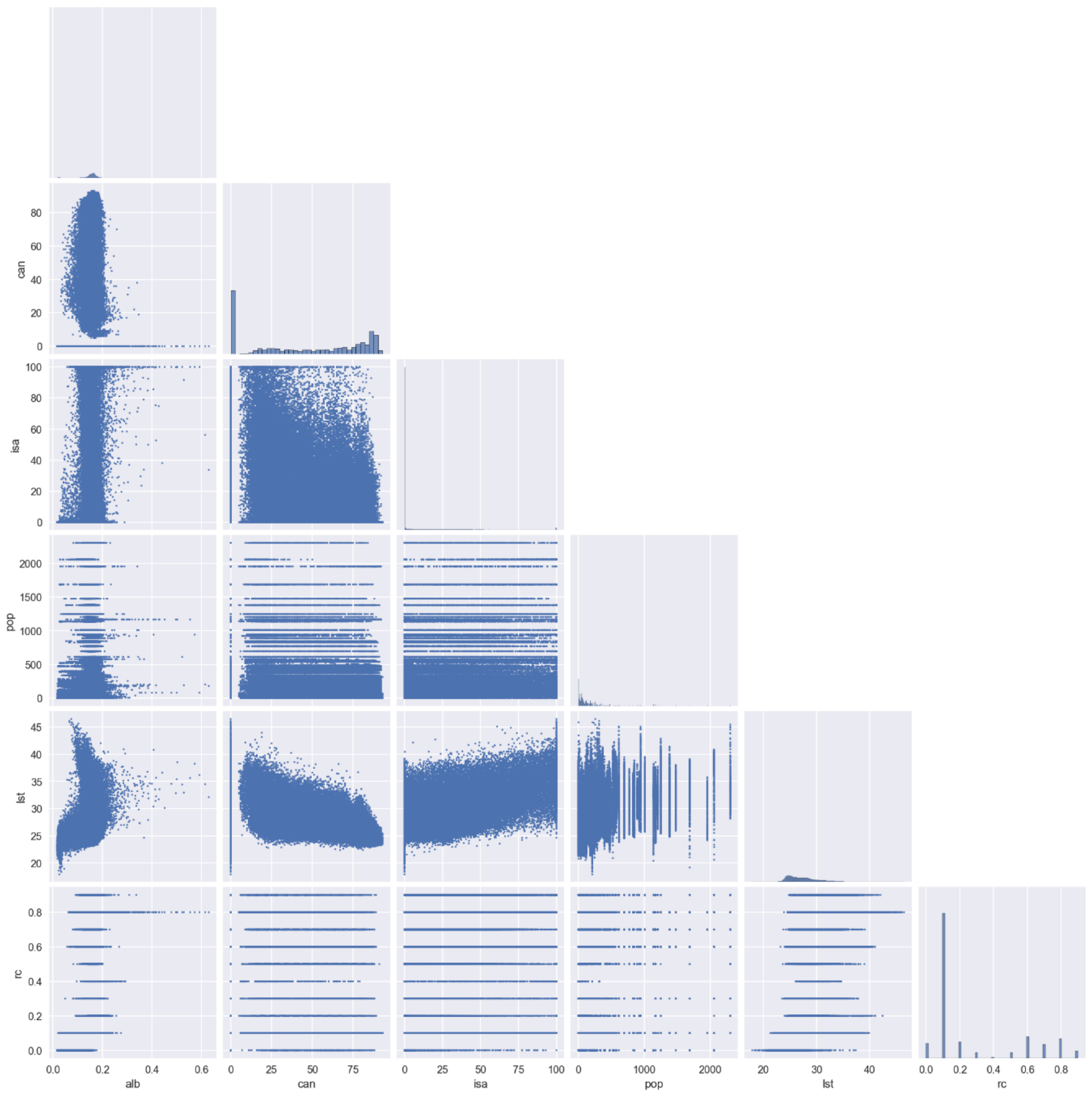


Figure 2-2 Pairwise scatter plot of the variables

2.2 Support Vector Machine (SVM)

Support vector machines (SVMs) are based on statistical learning theory and have the aim of determining the location of decision boundaries that produce the optimal separation of classes. It can be used for classification, regression and outliers detection in supervised learning problems. It works well in high dimensional spaces and is versatile as different Kernel functions or custom kernels can be specified for the decision function. (scikit-learn developers, 2021)

The model was built using the **scikit-learn** package. The data were preprocessed to have a mean of 0 and a standard deviation of 1, and were then passed into the **SVR** model. To fine-tune the hyperparameters, a grid of parameters were defined and the best set of hyperparameters was searched using the GridSearchCV method.

```python
from sklearn.svm import SVR
from sklearn.pipeline import make_pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import GridSearchCV

# instantiate a pipeline
regr = make_pipeline(StandardScaler(), SVR())

# defining parameter range
param_grid = {'svr__C': [0.1, 1, 10, 100, 1000],
              'svr__gamma': [10, 1, 0.1, 0.01, 0.001, 0.0001],
              'svr__kernel': ['rbf']}

grid = GridSearchCV(regr, param_grid, refit = True, verbose = 3)

# fitting the model for grid search
grid.fit(X_train, y_train)
```

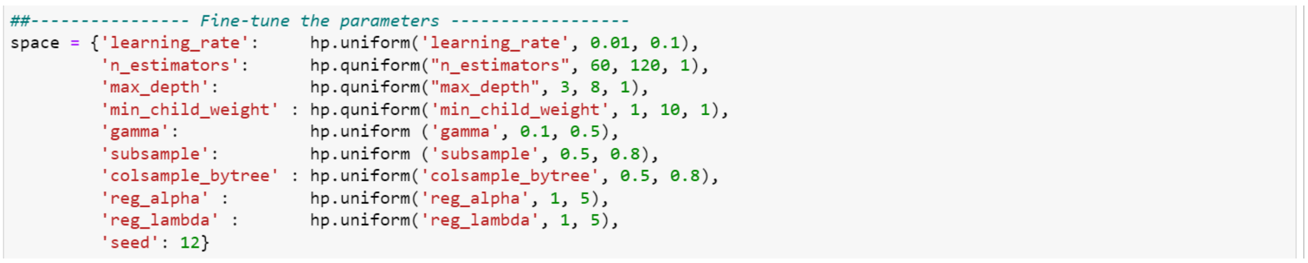
It should be noted that, a quadratic programming problem (QP) is at the heart of an SVM, which separates support vectors from the rest of the training data. The QP solver used by the libsvm-based implementation scales between $O(n_{features} \times n_{samples}^2)$ and $O(n_{features} \times n_{samples}^3)$, depending on how well the libsvm cache is utilized in practice (scikit-learn developers, 2021). That said, the computation complexity can increase dramatically with the increasing number of samples, which was observed during the training process. To reduce the training time, a random subset of 200000 data samples (out of 2638320 samples) were used for training model, this number should be sufficient to represent the full dataset well.

As an attempt to reduce the training time, a **LinearSVR** model was also built and trained. However, since it only uses a linear kernel, which could not be the case for this dataset, the model accuracy was not as good as the **SVR** model.

2.3 XGBoost

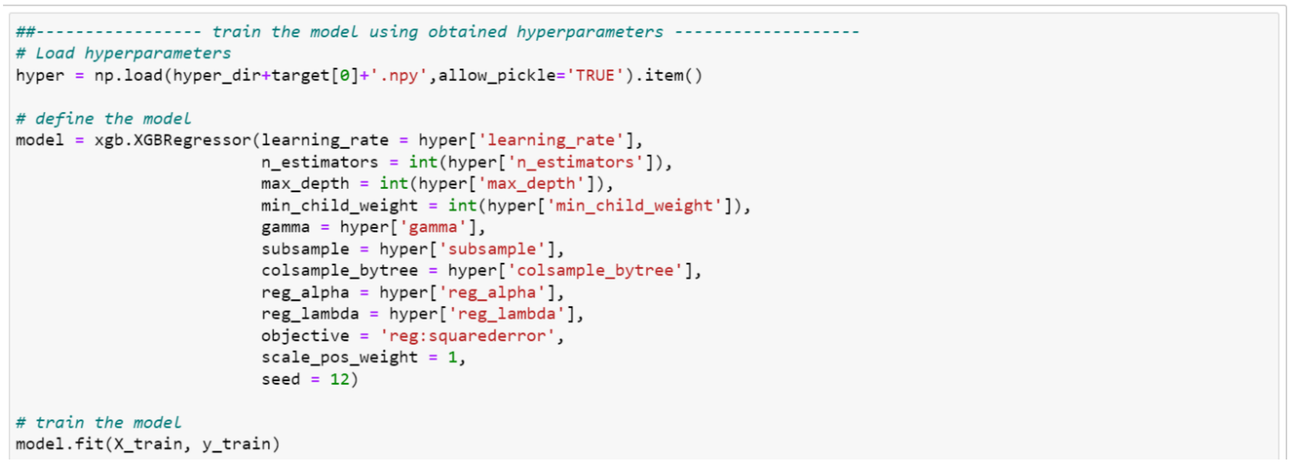
XGBoost is a decision tree based gradient boosting algorithm, which has been very popular for structured or tabular data. It offers parallel tree boosting (also known as GBDT, GBM) to solve a wide range of data science problems quickly and accurately. The same code can solve problems with billions of examples in major distributed environments (Hadoop, SGE, MPI). (xgboost developers, 2021) Since the dataset used in this study is structured and has more than 2 million samples, XGBoost could be a suitable choice to consider.

The model was built using the **xgboost** library. A range of hyperparameters were defined and they were fine tuned using the **hyperopt** library.



```
##--------------- Fine-tune the parameters -----------------
space = {'learning_rate':     hp.uniform('learning_rate', 0.01, 0.1),
         'n_estimators':      hp.quniform("n_estimators", 60, 120, 1),
         'max_depth':         hp.quniform("max_depth", 3, 8, 1),
         'min_child_weight' : hp.quniform('min_child_weight', 1, 10, 1),
         'gamma':             hp.uniform ('gamma', 0.1, 0.5),
         'subsample':         hp.uniform ('subsample', 0.5, 0.8),
         'colsample_bytree' : hp.uniform('colsample_bytree', 0.5, 0.8),
         'reg_alpha' :        hp.uniform('reg_alpha', 1, 5),
         'reg_lambda' :       hp.uniform('reg_lambda', 1, 5),
         'seed': 12}
```

After the best performed set of hyperparameters was obtained, they were passed into the XGBoost model. Then the model was trained using the full dataset, which has 2638320 samples.



```
##---------------- train the model using obtained hyperparameters ------------------
# Load hyperparameters
hyper = np.load(hyper_dir+target[0]+'.npy',allow_pickle='TRUE').item()

# define the model
model = xgb.XGBRegressor(learning_rate = hyper['learning_rate'],
                         n_estimators = int(hyper['n_estimators']),
                         max_depth = int(hyper['max_depth']),
                         min_child_weight = int(hyper['min_child_weight']),
                         gamma = hyper['gamma'],
                         subsample = hyper['subsample'],
                         colsample_bytree = hyper['colsample_bytree'],
                         reg_alpha = hyper['reg_alpha'],
                         reg_lambda = hyper['reg_lambda'],
                         objective = 'reg:squarederror',
                         scale_pos_weight = 1,
                         seed = 12)

# train the model
model.fit(X_train, y_train)
```

3. Results and Discussion

For the support vector regression (SVR) model, as shown in Figure 3-1, the best performed model achieved a training R^2 of 0.84 and a same testing R^2 of 0.84.

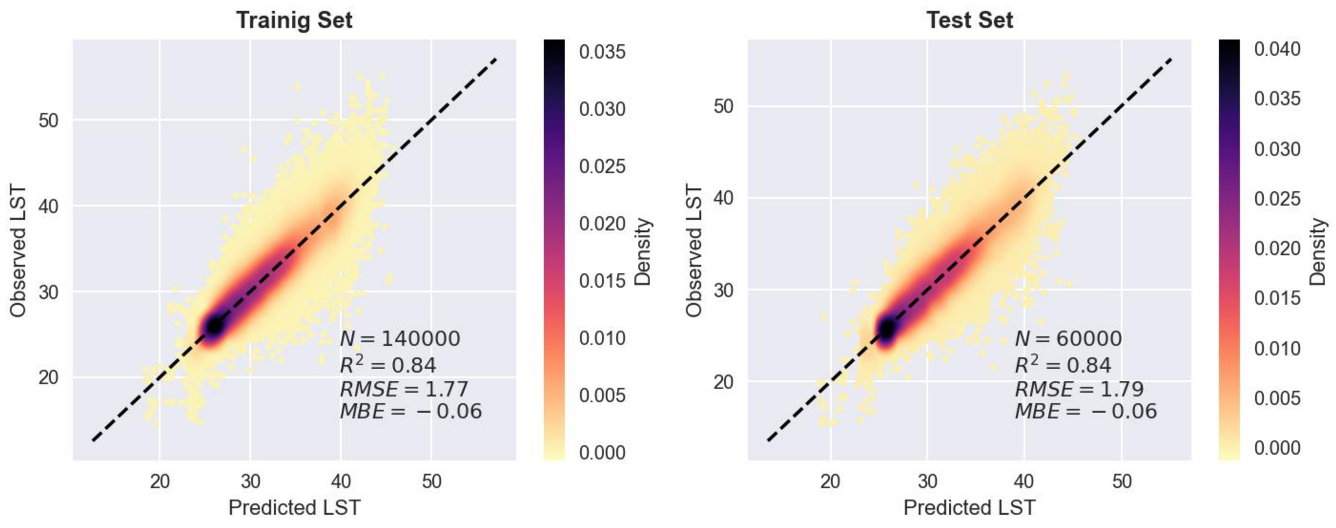


Figure 3-1 Training and testing results of the best performed SVR model

For the XGBoost model, in contrast to the SVR model, where the LUfull variable was replaced by its corresponding runoff coefficient values, the LUfull variable was directly handled as a categorical variable, as allowed by tree-based models. The best performed XGBoost model achieved a training R^2 of 0.87 and a same testing R^2 of 0.87 (Figure 3-2).

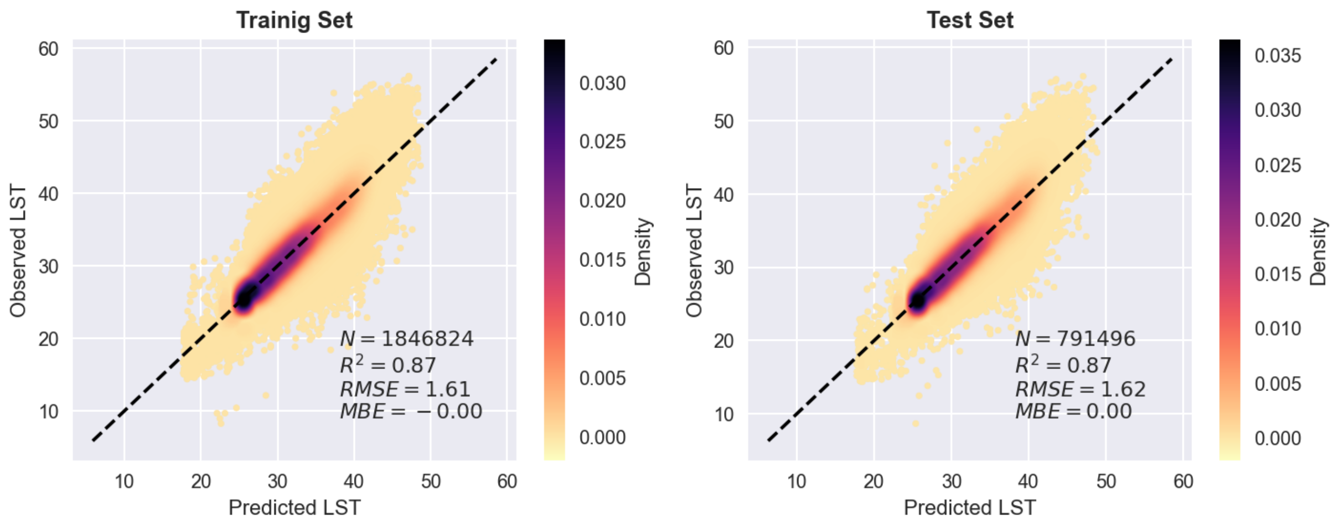


Figure 3-2 Training and testing results of the best performed XGBoost model

One advantage of the tree based models is that there are ways to compute the feature importance, so the model is more explainable. Using the **shap** library , which is a game theoretic approach to explain the output of machine learning models, the feature importance of the XGBoost model was calculated. As shown in Figure 3-3, tree canopy fraction (can) was identified as the most influential factor, followed by land use/land cover (LUfull), impervious surface fraction (isa), population density (pop) and albedo (alb).

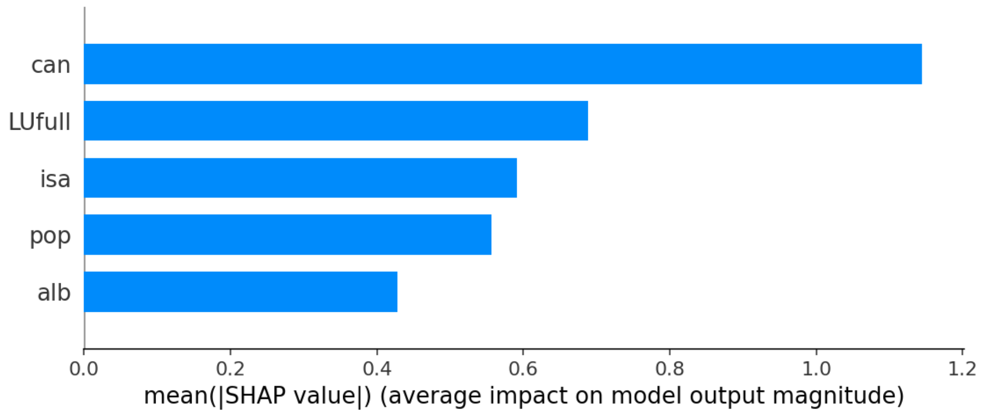


Figure 3-3 SHAP value for the predictors in the XGBoost model

The impact of each variable on the model output is shown in Figure 3-4. It was found that tree canopy fraction (can) was negatively correlated with land surface temperature (lst), whereas impervious surface fraction (isa) and population density is positively correlated with land surface temperature (lst).

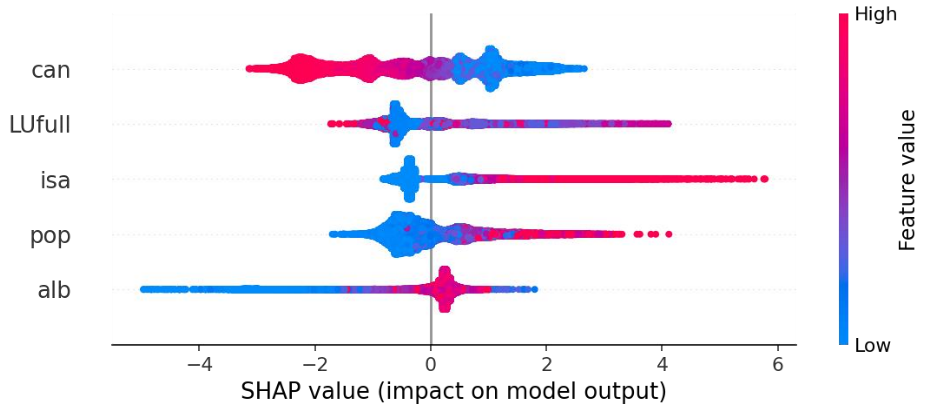


Figure 3-4 Impact of the predictors on the XGBoost model output

In general, the performance of these two models were comparable, with the XGBoost model having a higher testing R^2 of 0.87. The use of the runoff coefficient values in the SVR model improved the model performance, as compared to a model trained without the runoff coefficient variable. However, since there are over thirty types of landcover in the dataset, the accurate assignment of the runoff coefficient values could be unachievable, some bias could be introduced at that step.

As for the training speed, the XGBoost used the full dataset of more than 2 million samples and was still much faster than the SVR, which used a subset of 200000 samples. This should be attributed to a set of optimizations carried out for the XGBoost, including sparse-aware implementation, parallelization of tree construction, cache optimization, etc. (Chen & Guestrin, 2016)

4. Conclusion

To predict land surface temperature from a set of landscape metrics, two types of machine learning models were developed, using support vector machine (SVM) and XGBoost, respectively. Both models performed reasonably well in testing, with the XGBoost achieving a testing R^2 of 0.87, which is higher than the SVR model's 0.84. Because of a series of optimization designed for the XGBoost, it is also much faster than the SVR model.

Tree canopy fraction (can) was identified as the most influential factor to land surface temperature (lst) among the five independent variables, followed by land use/land cover (LUfull), impervious surface fraction (isa), population density (pop), and albedo (alb). In addition, tree canopy fraction (can) was found to be negatively correlated with land surface temperature (lst), whereas impervious surface fraction (isa) and population density were found to be positively correlated with land surface temperature (lst).

References

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785

Debbage, N., & Shepherd, J. M. (2015). The urban heat island effect and city contiguity. *Computers, Environment and Urban Systems*, *54*, 181–194. https://doi.org/10.1016/j.compenvurbsys.2015.08.002

Galletti, C. S., Li, X., & Connors, J. P. (2019). Establishing the relationship between urban land-cover configuration and night time land-surface temperature using spatial regression. *International Journal of Remote Sensing*, *40*(17), 6752–6774. https://doi.org/10.1080/01431161.2019.1594432

Kleerekoper, L., van Esch, M., & Salcedo, T. B. (2012). How to make a city climate-proof, addressing the urban heat island effect. *Resources, Conservation and Recycling*, *64*, 30–38. https://doi.org/10.1016/j.resconrec.2011.06.004

Lu, L., Weng, Q., Xiao, D., Guo, H., Li, Q., & Hui, W. (2020). Spatiotemporal Variation of Surface Urban Heat Islands in Relation to Land Cover Composition and Configuration: A Multi-Scale Case Study of Xi’an, China. *Remote Sensing*, *12*(17), 2713. https://doi.org/10.3390/rs12172713

NYC DEP. (2012). *Guidelines for the Design and Construction of Stormwater Management Systems*.

Nyman, D., Chiang, E., Eggleston, E., Eisenberg, B., Kennedy, M., & Maguire, T. (2002). *Hydrology Handbook for Conservation Commissioners: A Guide to Understanding Hydrologic and Hydraulic Data and Calculations under the Massachusetts Wetlands Protection Act*. Commonwealth of Massachusetts, Department of Environmental Protection, Division of Watershed Management, Wetlands and Waterways Program. https://archives.lib.state.ma.us/handle/2452/38865

scikit-learn developers. (2021). *1.4. Support Vector Machines*. Scikit-Learn. https://scikit-learn/stable/modules/svm.html

Shaker, R. R., Altman, Y., Deng, C., Vaz, E., & Forsythe, K. W. (2019). Investigating urban heat island through spatial analysis of New York City streetscapes. *Journal of Cleaner Production*, *233*, 972–992. https://doi.org/10.1016/j.jclepro.2019.05.389

Trlica, A. (2017). *Urban Land Cover and Urban Heat Island Effect Database* [Data set]. Harvard Dataverse. https://doi.org/10.7910/DVN/GLOJVA

xgboost developers. (2021). *XGBoost Documentation—Xgboost 1.5.1 documentation*. https://xgboost.readthedocs.io/en/stable/index.html

Zhou, W., Huang, G., & Cadenasso, M. L. (2011). Does spatial configuration matter? Understanding the effects of land cover pattern on land surface temperature in urban landscapes. *Landscape and Urban Planning*, *102*(1), 54–63. https://doi.org/10.1016/j.landurbplan.2011.03.009