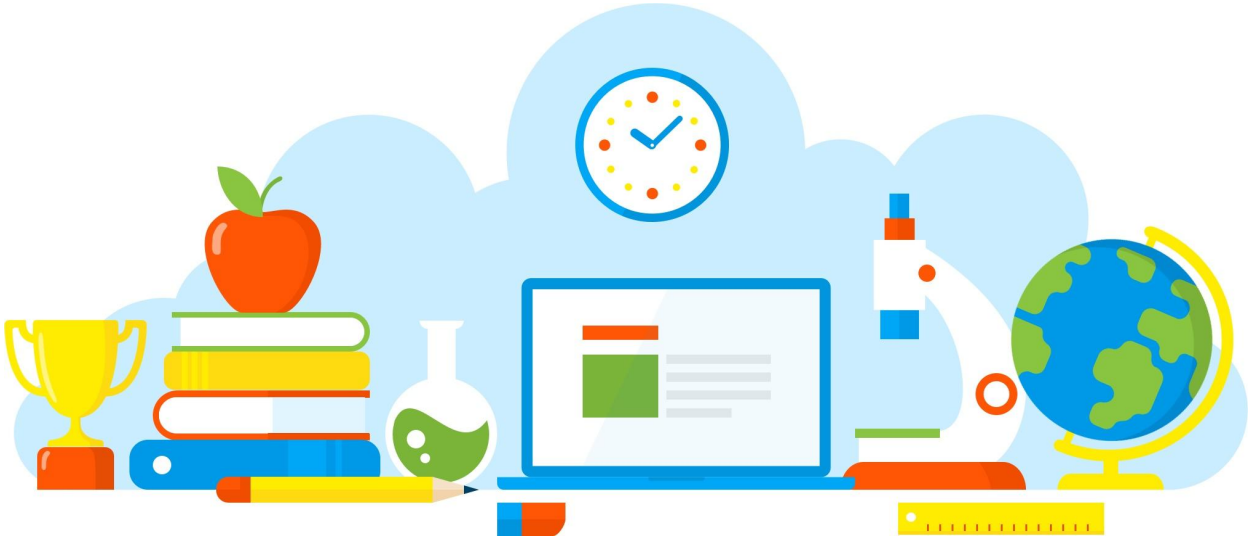


Predicting Historical Population Counts

with

Supervised Machine Learning

by Vitaly Vigasin

Introduction

This research employed data from the Seshat Databank (seshatdatabank.info) under Creative Commons Attribution Non-Commercial (CC By-NC SA) licensing.

The Seshat Databank is an instrument of a new and exciting field of scientific research called *cliodynamics* that applies mathematical methods to the research of the history of human societies. More on cliodynamics and the Seshat Databank can be found in the publications cited below:

Turchin, P., R. Brennan, T. E. Currie, K. Feeney, P. François, […] H. Whitehouse. 2015. "Seshat: The Global History Databank." Cliodynamics 6(1): 77-107. https://doi.org/10.21237/C7clio6127917.

Turchin, P., T. E. Currie, H. Whitehouse, P. François, K. Feeney, […] C. Spencer. 2017. "Quantitative historical analysis uncovers a single dimension of complexity that structures global variation in human social organization." PNAS. http://www.pnas.org/content/early/2017/12/20/1708800115.full.

The subject of this analysis is selecting and evaluating a model for predictions of human populations based on historical data available from the Seshat Databank.

We will use the *Equinox-2020* dataset available for download from the Seshat website which contains a big deal of historical information on various societies found in many geographical regions all over the world including population counts.

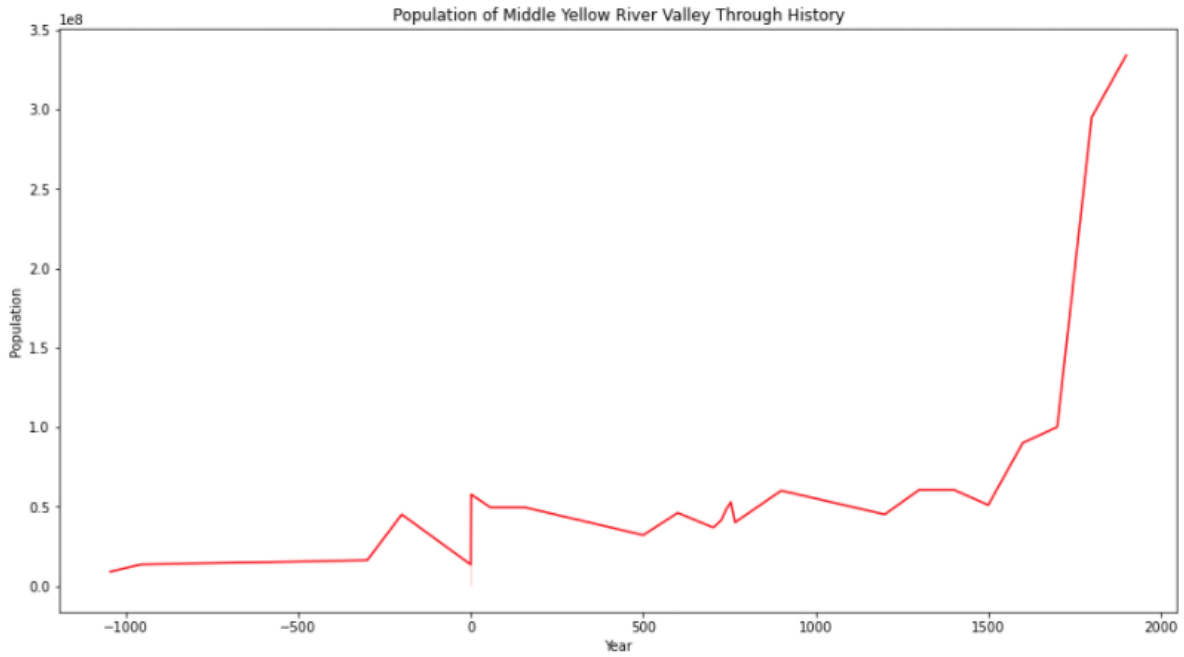
For the purpose of this exercise we will select the so-called National Geographic Region ("NGA" in Seshat's terminology) of Middle Yellow River Valley in the Henan province of China surrounding the city of Zhengzhou. This is the ancient centre of the Chinese civilization, and the Seshat Databank has population data for this region for several millennia.

Data Analysis

The data have been prepared and cleaned with the following steps:

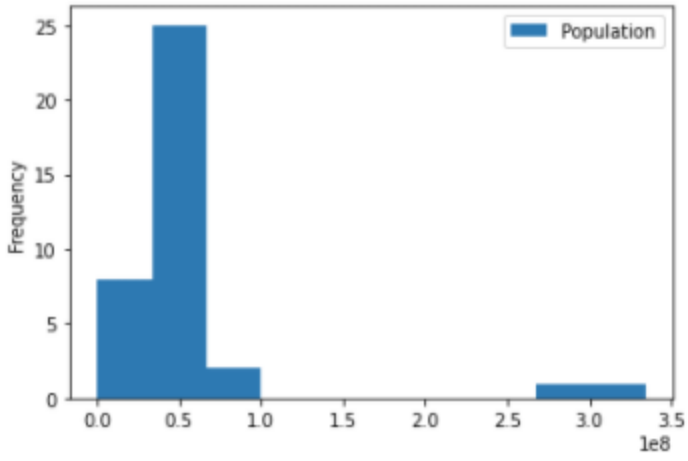
- Selecting only population-related data from the dataset
- Selecting rows that contain population data for the target region
- Filling in empty values
- Transforming dates so that they are negative and positive integers instead of strings containing ‘CE’ and ‘BCE’ parts denoting historical eras
- Replacing values that contain a range with its mean value

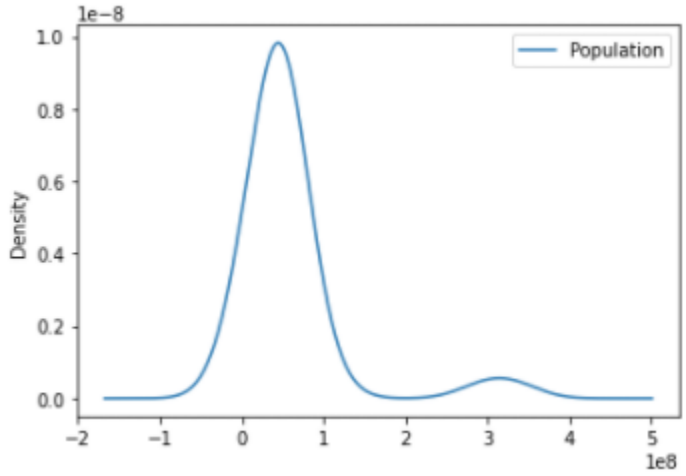
The whole process resulted in a new dataframe which used as a basis for the plot shown below:



The resulting dataframe consisted of 37 rows.

A probability density plot showed close to normal distribution with some outliers:





Additional attempts at further normalization using the Normalizer transformer did not result in an improvement of models performance and was not subsequently used. Normalization parameter was, however, enacted for the model object during its initiation.

Model Selection

Since we are dealing with a regression problem, the following regression models were selected for evaluation:

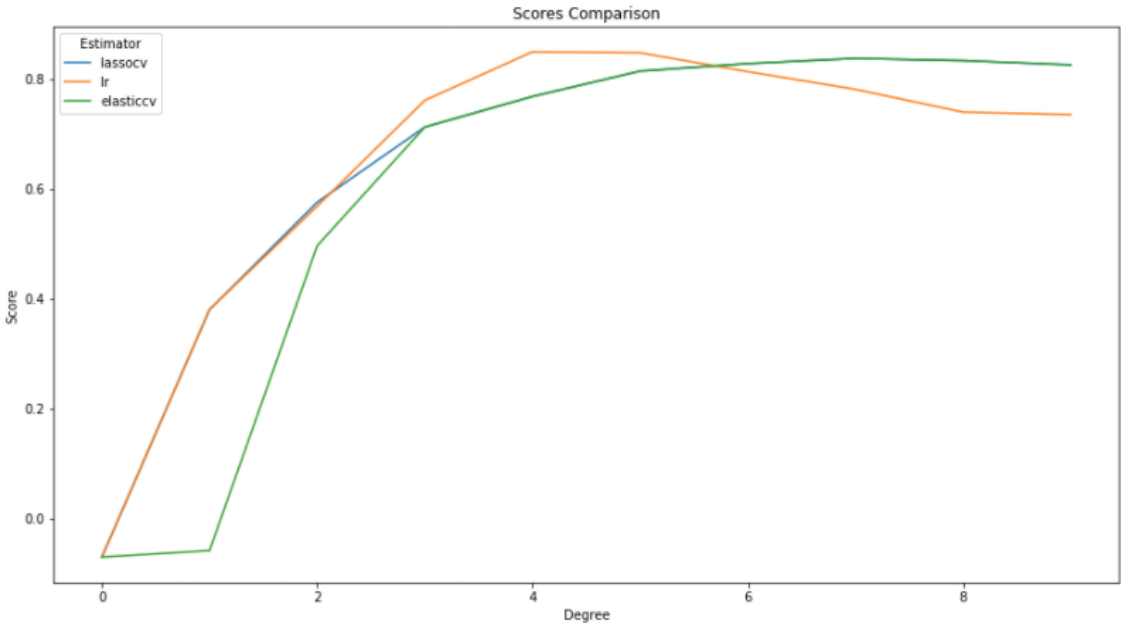
- classical Linear Regression
- RidgeCV
- LassoCV
- ElasticNetCV

A regression pipeline was created that employed polynomial features and a test was run for all of the above estimators while varying the degrees of polynomial regression.

Of the three, RidgeCV produced the worse results with a score lower than 0.6 and was subsequently removed from further consideration.

The remaining three estimators produced their best scores as shown below:

Estimator	Degree	Score
lr	4	0.848471
lr	5	0.846688
elasticcv	7	0.836564
lassocv	7	0.836564
elasticcv	8	0.832443



As we can see, the best score was achieved using vanilla LinearRegression with polynomial features of 4th degree.

However, tuning the parameters of its closest rival LassoCV as shown below allowed to improve the score and ultimately led to adopting LassoCV as the best model instead of the vanilla Linear:

make_pipeline(PolynomialFeatures(degree=8), LassoCV(normalize=True, cv=5, max_iter=10000))

Playing with parameters of ElasticNetCV did not produce a better score, and the algorithm was abandoned as the result.

It is worth mentioning that aforementioned the test was performed with a **training set** set at the 0.2 mark, and the best performance was demonstrated with the split no higher than that, which held true for all of the models under consideration:

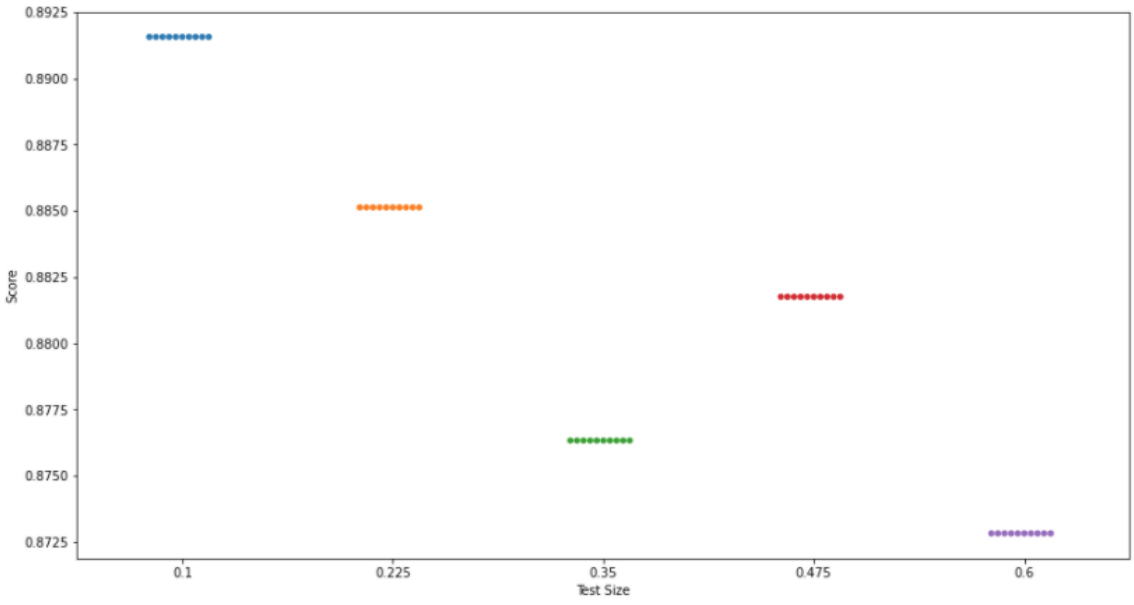
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

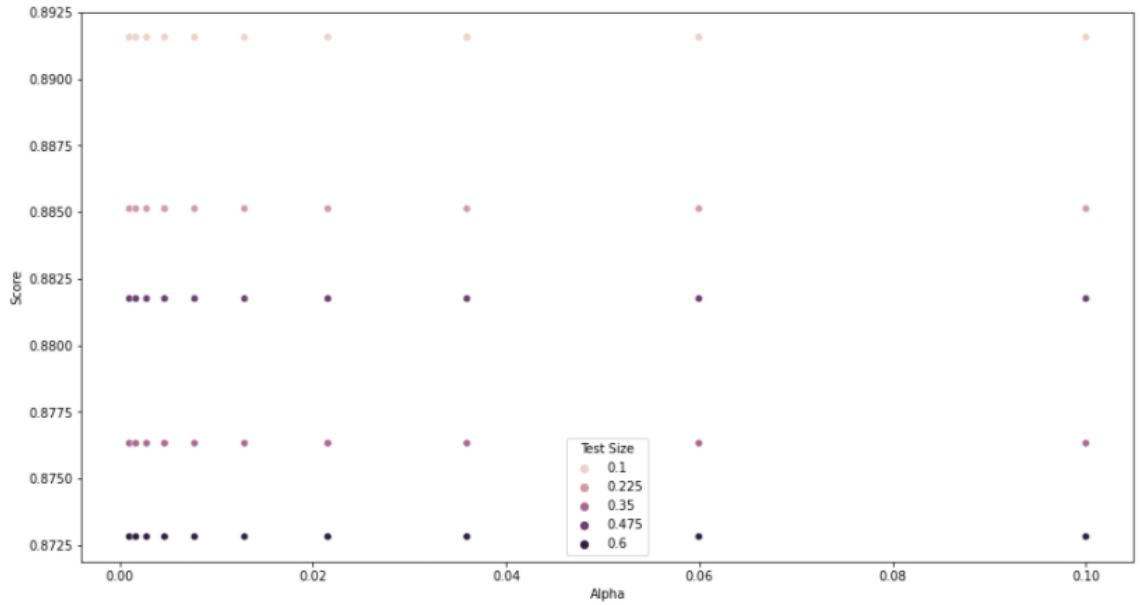
Regularization

An additional test was run for the sake of an attempt to further improve the score using the regularization technique. With this execution, the LassoCV model was tested with varying alphas split as shown below:

```
alphas = np.geomspace(0.001, 0.1, 10)
```

Additionally, varying levels of train set size were tested in the range from 0.1 to 0.5:





As we can see, for a chosen train set size the alpha parameter had no effect on model's accuracy and no further improvement in the score was achieved.

However, based on the result of this test, some improvement was found to be possible with a lower train set split, and the split parameter was set at 0.1 instead of 0.2 as used earlier.

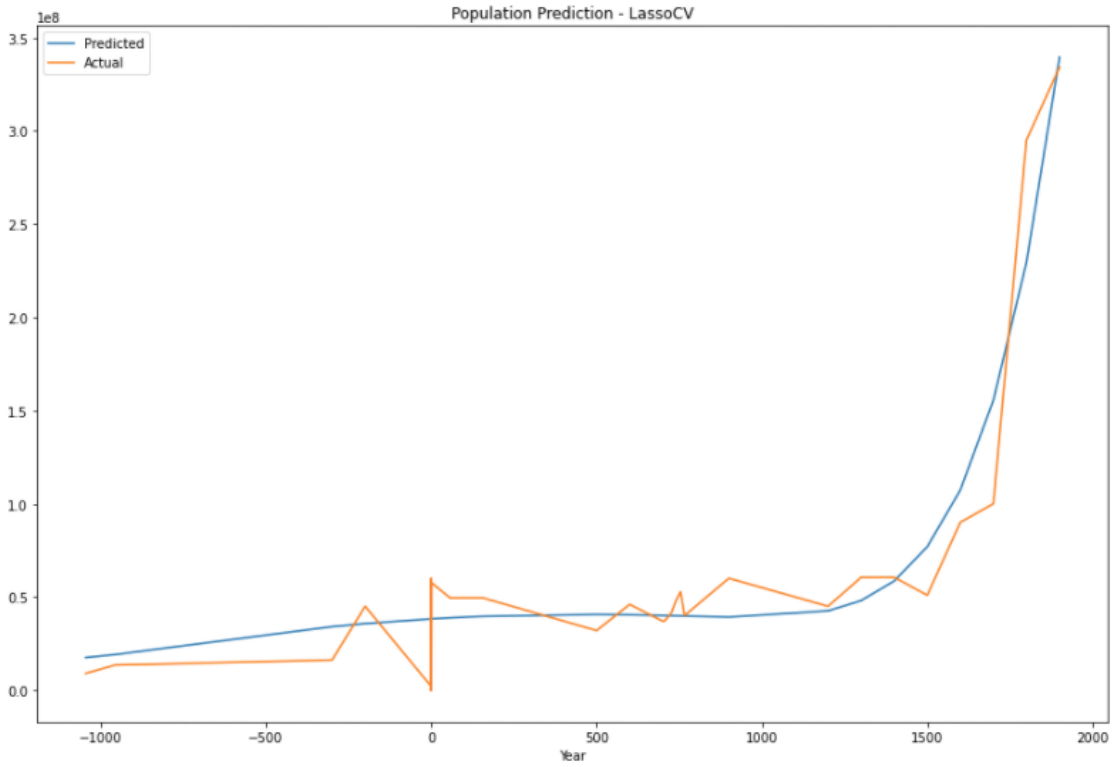
Final Model

The final model was selection was set at LassoCV to be run with the parameters as shown below resulting in a final test score of 0.8915651269911424 which was a significant improve of the previously highest score of 0.8660258677831475 owned by the vanilla LinearRegression:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.1, random_state=42)

pipe=make_pipeline(PolynomialFeatures(degree=8), LassoCV(normalize=True, cv=5,
max_iter=100000)).fit(X_train, y_train)
```

The final prediction along with the ground truth is shown below:



Conclusion

As we can see from the last plot, the selected model and its parameters produces a very good approximation of the true values and can indeed be used as a good predictor for the purposes of this research.

The fact that the model is resilient to the variance of the ground truth bears special mention.

As the population category having been considered in this paper is just one of the about 30 others representing various areas of the world as represented in the Seshat Databank, it is of interest to test the same approach for the rest of the available population data.