Regression techniques to predict violent crime based on socio-economic factors

An De Rijdt[1]

[1] VU Amsterdam
a.j.de.rijdt@student.vu.nl

Abstract. In this paper we describe exploration of the communities and crime dataset. We investigate different models to predict violent crime rate based on socio economic variables. Best performing model is a random forest applied after standardization and normalization of the feature space. The analysis was performed using python in a jupyter notebook.[1]

Keywords: Data mining, Crime, Regression

1 Data exploration, cleaning and feature engineering

We use the UCI Machine Learning dataset [2] for our analysis. This contains 2215,samples and 147 features. We want to model the violence per capita (violentPerPop) variable. We notice that before starting to apply regression models we have to cleanup the data, extract meaningful features and deal with missing values.

A lot of samples are missing our target variable. From information of the dataset [2] we know that the target variable is calculated by population and sum of murder, rape, robbery, and assault. This could be used to impute the missing values. However for most of those the rapes data is missing. For the remainder the assault data is missing so we so we decide to drop all samples with missing violentPerPop.

We can now drop all specific crimes. As we explained they are all summing up to the violentPerPop. In fact they could be target variables themselves, not predictors.
We also drop the column indicating the number of non violent crimes as we only want to perdict the violent ones on the basis of socio economic variables.

Communityname contains the type of community (city, township, etc), which might be a strong indicator so we engineer a categorical feature “communityType” from this after which we drop the communityname feature.

For the columns that have more than half of the values missing we create a new feature indicating whether it is missing or not and drop the original column. The only other remaining missing value for otherPerCap is imputed by the median.

2 Correlations and feature selection

We explore the data visually and look at meaningful correlations. We notice that a lot of variables have an exponential distribution, including the target variable.

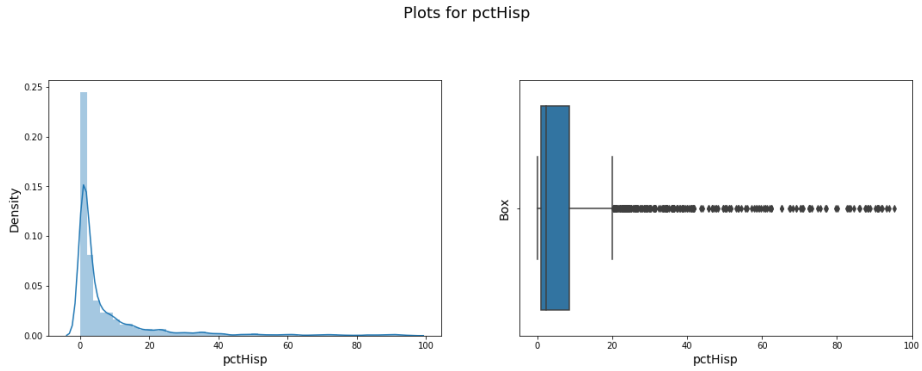

Fig. 1. A lot of variables have exponential distribution

We make a simple countplot for each categorical variable to see how the records are distributed over the different categories. We see that the data is not sampled evenly over states and over community types (way more data points from cities).

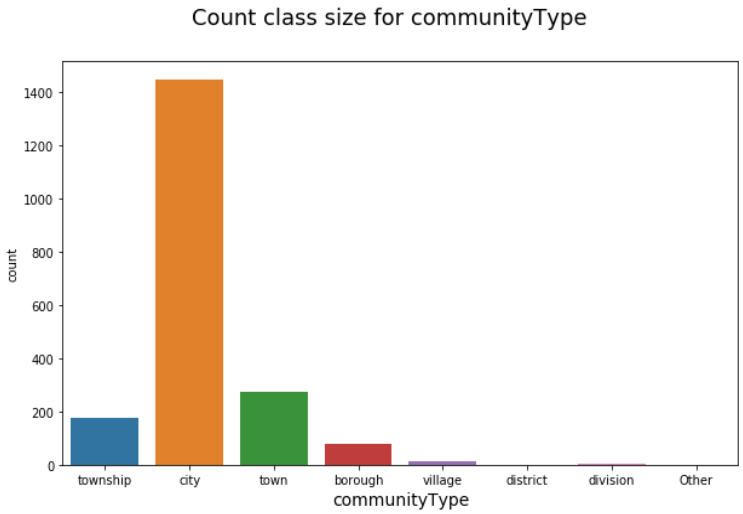

Fig. 2. Most communities are cities

Looking at correlation between features, we notice there are many highly correlated features. We set the treshhold to 0.80 and keep only one for all features that have correlation higher than this treshhold. We keep 53 features this way.

In below table we see that some features have very high correlation with the target variable. We will see later they also appear as most important features in our models.

Table 1.

pct2Par	0.701390
pctWdiv	0.631025
pctMaleDivorc	0.585157
pctPersOwnOccup	0.570882
pctPubAsst	0.561305
pctBlack	0.533936
pctSmallHousUnits	0.529933
pctPoverty	0.522133
medIncome	0.480801
medNumBedrm	0.409800
pctLowEdu	0.400825
communityCode_missing	0.388535

3 Regression

We encode by dummy variables rather than one hot encoding to avoid strong collinearity affecting the models. We set up a scikit-learn pipeline both with and without PCA as dimensionality reduction technique. Further preprocessing steps are standardization and normalization as many variables are exponentially distributed. We use a train-testsplit of 50/50 for training and validation. We compared following regression algorithms: simple regression, penalized regression, Random Forest, Neural network, KNN with different numbers of neighbours, XGBoost.

3.1 Some general observations

We mentioned earlier that the target variable had exponential distribution. When applying above algorithms without transformation, we do not get very good results (large errors and lower r squared). Hence we decide to log transform the target variable (with a correction to avoid log(0)) which indeed improved model performance.

Given the large amount of remaining features, we decided to also try running the models with a limited feature set (selecting on the basis of correlation with the target variable) but this did not improve our models.

Applying PCA as a preprocessing step however improved crossvalidation score of linear regression and also the test error, which is not surprising as it decorrelates the features and will prevent overfitting. It does not improve random forest (r squared, train and test error are all worse), which is also expected as random forest already performs good regularization.

3.2 Validation of different models

In table 2 some of the model results are compared. We see that Random forest is the best performing model, both on mean square errors on train and testset and on r squared. Overall models generalize well, test error is similar to train error.

Table 2. Model results

	test error	**train error**	**R2**	**test error no pca**	**train error no pca**	**r squared no pca**
Linear Regression	0.34	0.39	0.68	0.38	0.36	0.72
Neural Net	0.27	0.30	0.76	0.30	0.31	0.76
Random Forest	0.09	0.08	0.93	0.06	0.06	0.95
XGBoost	0.19	0.19	0.85	0.19	0.17	0.87

3.3 Grid search on random forest

To tune the random forest we perform grid search of the parameters, resulting to best random forest with: minimum samples in a leaf 3, 1000 estimators. Model feature importances are as expected from the correlation analysis earlier.

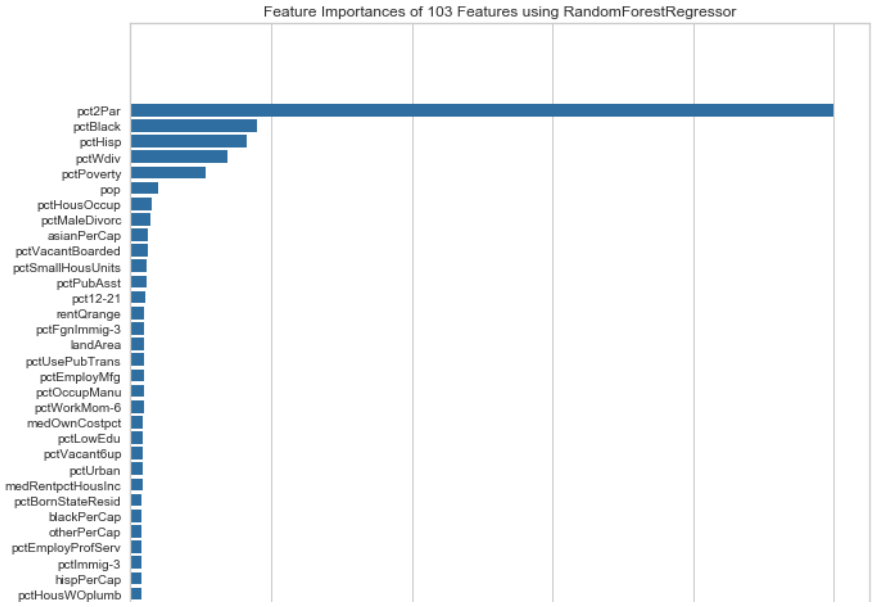

Fig. 3. Percentage families with two parents is the most important feature in the random forest model.

References

1. An De Rijdt: https://github.com/klimantje/DataMining/blob/master/Crime%20prediction.ipynb
2. Dua, D. and Karra Taniskidou, E. (2017). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml/datasets/communities+and+crime]. Irvine, CA: University of California, School of Information and Computer Science.