Predictive Modeling of California Electricity Billing Data

Liu Yiming

University of California, Irvine

June 2017

1 Actual vs Imputed

From the side by side histograms of Figure 1, we can note, in general imputed data seem to represent the actual billing data reasonably well, except that the actual billing data seem to exhibit a bit heavier tail, which might in turn implicate the imputed data might not be able to precisely represent the edge cases.

2 Data Preprocessing

The starting condition of the given data set is relatively poor. Thus various techniques were employed to clean up the data beforehand. One important thing to note is that all analysis presented in this report was conducted on the data-set with all features on the remove list removed. The clean up procedure I used for this specific data set is as follows.

1. Randomly shuffled the data set
2. Removed all features which have more than 5 percent of its entries being NA
3. Imputed the missing values with corresponding medians
4. All numerical features were then scaled by a Min-Max scaler, i.e. subtracted the minimum value and divided by Max value minus the minimum value
5. Categorical features used in the analysis were one-hot-encoded, i.e creating dummy features for each level of the used categorical features

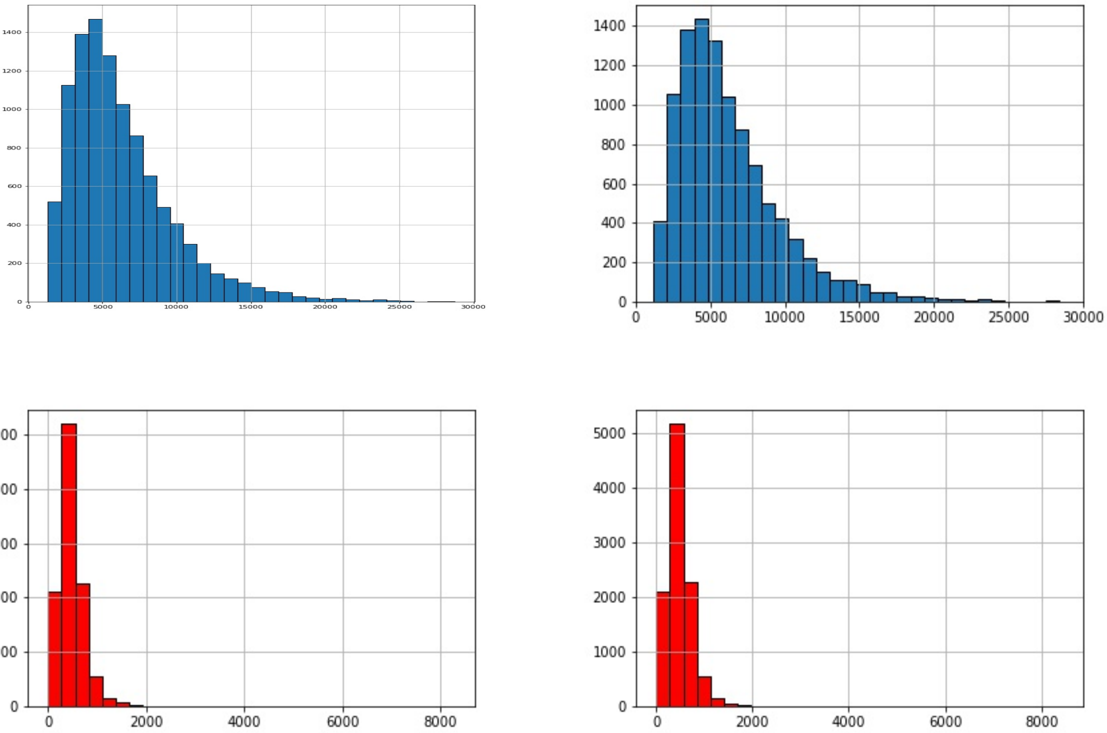

Figure 1: Side-by-side histograms of imputed and actual billing data. Top from left to right:kwh-frommonthly and ann-kwh. Bottom from left to right:ng-frommonthly and ann-therm

3 Methods

Selecting the best model for this dataset is actually a pretty tricky thing, since firstly we had limited knowledge over the actual meanings of some of the features, secondly lots of numerical features according to the documents provided are actually label encoded categorical features with more than two levels, i.e categorical features with levels represented by a numerical number. Thus parametric models such as linear regression and support vector machine might be prone to unwanted bias, if these parameters are not handled properly. Although the above mentioned problem can be alleviated by simply one-hot encoding those features, given the limited number of observations, we might not wish to induce further sparseness into the dataset. Therefore, given above thoughts, Random-forest and Gradient Boosted decision tress(GBT) seemed to be promising algorithms in this case, since they can quickly adapt to the structure of the dataset, and not restricted to pre-specified parametric form.

3.1 Random Forest and Bagging

Random-forest is built upon the idea of bagging(bootsrap aggregation), which is a general ensemble method like boosting, and not restricted to just decision trees. Random-forest is designed with the purpose to alleviate the over-fitting tendency of simple decision trees, by training multiple decision trees on bootstrap sampled observations and a randomly selected subset of features. Thus as a matter of fact, each decision tree is only able to see a part of the structure of the dataset, and only able to overfit part of the data. Hence when we average over all the trees with respect to the entire dataset, we are able to effectively reduce model variance, leading to better generalization. In essence, bagging is a way to reduce variance of overfitting estimators

3.2 Boosting

Boosting as mentioned previously is another important ensemble method. Here, I will only briefly discuss a specific boosting method, gradient boosting, since it's relevant to our analysis. the idea of gradient boosting is to train learners sequentially upon the residual errors of previous learner. Unlike bagging, gradient boosting is a sequential process while bagging is parallel. Another major difference is gradient boosting is in fact trying to reduce bias sequentially. Thus, gradient boosting is prone to overfitting, which is exactly what we try to reduce when bagging.

3.3 Modeling Strategy

By combining the features of both bagging and boosting, The general modeling strategy used in this analysis is as follows

- Train a random forest learner as a baseline learner
- Tune the baseline random forest learner with randomized search over a great variety of possible parameter settings
- **In Parallel**
- Train a GBT learner as a baseline learner
- Tune the baseline GBT learner with randomized search over a great variety of possible parameter settings
- **After both independently learned learners are tuned**
- Select the best performed learner as a base learner
- Train a bagged learner, which consists of multiple best base learners chosen in last step, as the final learner

Comparing above method to the method described in the documents provided, there are a few major discrepancies. Firstly, our basic building block decision tree is essentially non-parametric, while linear regression used in original document is parametric. Secondly,our final learner is an ensemble of complex learners, while their final learner is a single complex learner, which might be prone to overfitting.

4 Results

4.1 KWH

Table 1: Cross validation MSE on each fold and mean MSE

	Fold-I	Fold-II	Fold-III	Fold-IV	Fold-V	Mean
RT	4288602	4683398	4342491	4267412	4172777	4350936
GBT	3856538	4506151	4049051	3921770	4067138	4080130
BAG	3731000	4353247	3930043	3887136	3901624	3960610

Parameter settings of the final model

- RT(Random Forest): Max-depth=29, Max-features=350, Number-of-trees=300, Min-sample-leaf=5
- GBT(Gradient Boosted Trees): Max-depth=10, Max-features=100, learning-rate=0.1, Number-of-trees=75, Min-sample-leaf=5
- BAG is a bgging of 10 GBT learners described above

4.2 NG

Table 2: Cross validation MSE on each fold and mean MSE

	Fold-I	Fold-II	Fold-III	Fold-IV	Fold-V	Mean
RT	47191	73852	41568	44987	59896	53499
GBT	45847	70093	38725	41742	57173	50716
BAG	44495	70983	37909	40507	55697	49918

Parameter settings of the final model

- RT(Random Forest): Max-depth=28, Max-features=300, Number-of-trees=300, Min-sample-leaf=5

- GBT(Gradient Boosted Trees): Max-depth=5, Max-features=170, learning-rate=0.1, Number-of-trees=150, Min-sample-leaf=8

- BAG is a bgging of 10 GBT learners described above

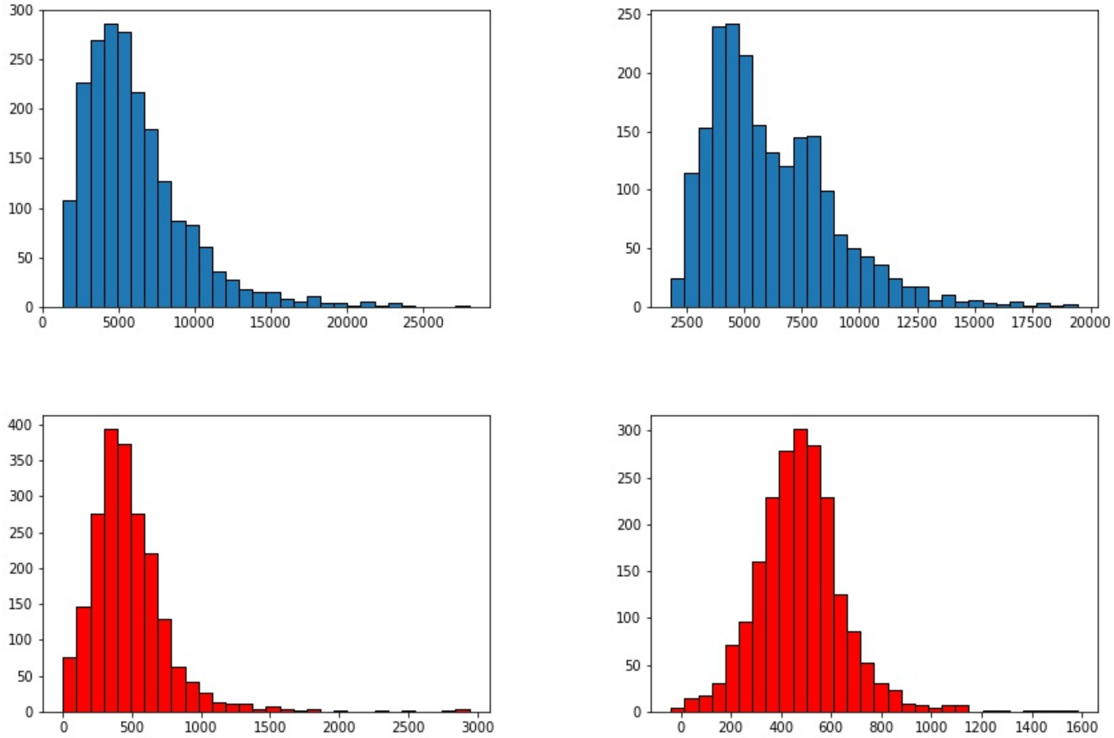

Figure 2: Above histograms compare the distribution of predictions made by our final models to actual actual distribution of kwh and ng based on a separate test set, that consists of 20 percent of total available observations, extracted before training. Top from left to right:kwh-frommonthly and predicted kwh. Bottom from left to right:ng-frommonthly and predicted ng