

DS 5110 Project Report

Exploratory and Predictive Analysis of Big Cities Health Coalitions Dataset

Kunjan Khatri, Nikson Panigrahi, Prakhar Patidar, Rishabh Shanbhag

Northeastern University, Boston, Massachusetts

1 Summary

The data set is based on the Big Cities Health Inventory (BCHI) [1] which is an open data platform managed by Big Cities Health Coalition (BCHC)[1]. It contains a collection of data representing a snapshot from 30 large cities in the United States of America. The objective of this project is to gain insights from the health dataset, find out which diseases are more prevalent and major cause of mortality in urban environment, raise awareness about certain diseases which are not familiar to general public, try to predict which disease a person is likely to have given a set of variables like city, age, sex.

The dataset is available in .csv format containing 15 columns/variables which includes indicator category, indicator, age, year, sex, race/ethnicity, value, place, etc. The categories of diseases are merged in observations and hence will require tidying up the data a little bit to create variables of interest. Apart from this a certain number of data pre-processing steps are required to clean up the data and produce insights from it. Consequently, it is planned to produce various insights from the data in the form of visualizations and find correlations and other patterns in the data, to help in modelling and produce interesting results. Identify most prevalent disease category using various descriptive and predictive analytics-based approaches. Applied machine learning models to predict whether the disease is present or not. We explored and compared several predictive models, including linear models and a gradient boosting method. The process revealed the strengths and weaknesses of each approach in terms of implementation complexity and predictive value. We used positive likelihood ratio in addition to accuracy as a performance metric to evaluate our models.

2 Methods

2.1 Data Pre-processing

The data set was downloaded from the BCHC [1] website and loaded into data frame using the tidy-verse package

as *BCHI.data*. The Place column containing data locations was separated into 3 new columns City, County and State which to make location-based visualizations feasible. Location coordinates were generated applying **geocode** function on City from **ggmap** [6] library making calls to Google maps API. The data frame containing city coordinates was exported as a csv file for future use.

Using the tidy verse package, we then applied filtering to clean the data and adjust for the problematic missing values making it more suitable for exploratory analysis of the Indicator Categories and Indicators.

After some EDA, we found out that the *BCHI.data* was mostly categorical and not suitable for predictive analysis. Using the result from EDA, Heart Mortality Rate was found out to be appropriate predictor to be used for predictive analysis. The HeartUCI [2] data set to provide a more precise and detailed representation of factors affecting Heart Mortality rate. This data set was joined with the *BCHI.data* using Indicator as a key.

City coordinates data was joined to the *BCHI.data* data frame containing all the data using City as the key so that each data point was mapped to latitudinal and longitudinal data.

2.2 Exploratory Data Analysis

In order to derive meaningful insights and study possible patterns, we started our analysis by coming up with inquisitive questions, answers to which might help us gain a deeper understanding of our data set. The questions are:

1. How is the data set distributed among the participating states and cities in BCHC project?
 - Figures 1 and 2 along with a short description in the results section answer this.
2. Can we observe a trend for Indicator Categories over the years?
 - Figures 3 and 4 along with a short description in the results section answer this.

3. What is the distribution of Indicators under the prevalent Indicator category (Chronic Diseases), and which Indicator is prevalent among them?
 - Figures 5 and 6 along with a short description in the results section answer this.

2.3 Feature Identification

2.3.1 Correlation Matrix

Correlation matrices are one of the best ways to identify and select the best predictor variables, it defines the correlation between several variables and the best correlation with the target variable can be identified. The correlation can either be positive or negative.

2.3.2 K-fold cross validation

The K-fold cross validation is another strategy to select the best training data, it involves making k folds of the training and testing set and calculating the respective error in each fold. The fold with the best performance is then selected. In our case we did this to ensure we select the right set of portions of data from the partition for training and testing.

2.3.3 Stepwise model selection

The stepwise model selection is a great strategy to verify if the selected predictor variables are really good for the model, for this we used `ols_step_best_subset()` from `olsrr` [5] package. This automatically creates the combinations of variables and does stepwise model selection and generates graphs, which can automate the process of stepwise model selection.

2.4 Modelling

2.4.1 Linear Model

Linear model is one of the basic regression models in machine learning. We chose this procedure to simplify the process and easier analysis of the models. The Linear Model also served as a base to verify the best possible predictor variables selection, as residuals were generated from the model fit by linear model. For the purpose of this project the basic `lm()` function in R. Furthermore, it was expected that this simple yet effective method for modelling can yield satisfactory on the continuous variables from the dataset. Predictions were obtained from the test set and were compared with the actual values by generating confusion matrix.

2.4.2 LS-SVM

LS-SVM (Least Squared Support Vector Machines) was chosen as another modelling technique to train our model, LS-SVM though basic is known to perform very well in machine learning problems. The process involved

was like the process of Linear model, we used the training partition to train the model and testing partition to test the model. LS-SVM proved to be the best model in our analysis giving impressive results. For verification of the results we used confusion matrices, as they are easy to visualize the results and generate some detailed metrics for errors.

2.4.3 Random Forest

The random forest algorithm [4] works by aggregating the predictions made by multiple decision trees of varying depth. Every decision tree in the forest is trained on a subset of the dataset called the bootstrapped dataset. The portion of samples that were left out during the construction of each decision tree in the forest are referred to as the Out-Of-Bag (OOB) dataset. In random forest, however, we randomly select a predefined number of features as candidates.

When the random forest is used for classification and is presented with a new sample, the final prediction is made by taking most of the predictions made by each individual decision tree in the forest. In the event, it is used for regression and it is presented with a new sample, the final prediction is made by taking the average of the predictions made by each individual decision tree in the forest.

2.4.4 XGBoost

XGBoost (which stands for extreme Gradient Boosting) [3] is an especially efficient implementation of gradient boosting. XGBoost is an implementation of the Gradient Boosted Decision Trees algorithm. We go through cycles that repeatedly builds new models and combines them into an ensemble model. We start the cycle by taking an existing model and calculating the errors for each observation in the dataset. We then build a new model to predict these errors. We add predictions from this error-predicting model to the ensemble of models.. To make a prediction, we add the predictions from all previous models. We can use these predictions to calculate new errors, build the next model, and add it to the ensemble. In practice, the initial predictions can be naive. Even if its predictions are wildly inaccurate, subsequent additions to the ensemble will address those errors.

3 Results

3.1 Exploratory Data Analysis:

For Exploratory Data Analysis, the following questions were answered:

1. How is the data set distributed among the participating states and cities in BCHC project?

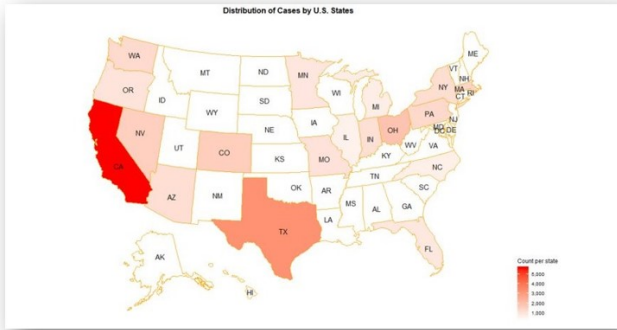


Figure 1: Distribution of Cases by US states

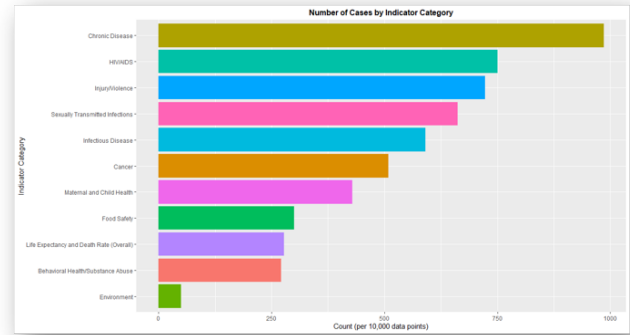


Figure 4: Number of Cases/Indicator



Figure 2: Distribution of Cases by US Cities/Indicator

- From Figure 1 and Figure 2, we could infer that California, Texas, and Ohio are the states with highest recorded cases for all the indicator categories. Diving in the cities we found that, Los Angeles has most of the cases reported for sexually transmitted diseases, Houston for AIDS diseases, and Cleveland for Chronic Diseases.

2. Can we observe a trend for Indicator Categories over the years?

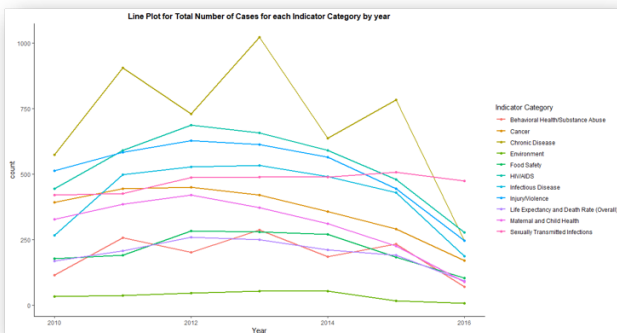


Figure 3: Total Cases for Indicator/Year

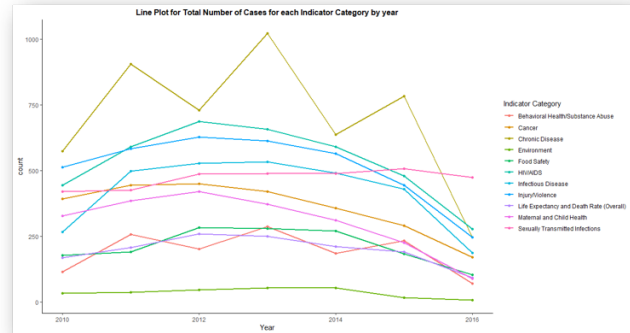


Figure 5: Total Cases for Indicator/Year

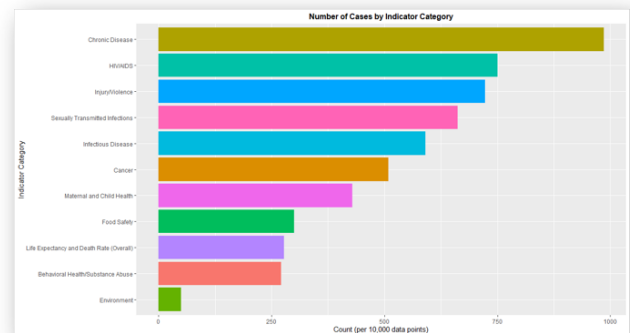


Figure 6: Number of Cases/Indicator

- From Figure 5, we see that Heart disease mortality rate tops among all the indicators for

Chronic Diseases, then from the Figure 6, we see that there is a significant decline in the recorded cases in the trend of all the chronic disease indicators over the years, which is unusual, and maybe a result of the data being skewed towards previous years as compared to recent years. This may have happened because of lower recent recordings of the cases by the BCHC.

- Thus, we used additional data set for heart disease to do CDA and PDA on Heart Disease Mortality rate (Prevalent Chronic Disease Indicator).

3.2 Feature Selection:

3.2.1 Correlation Matrix

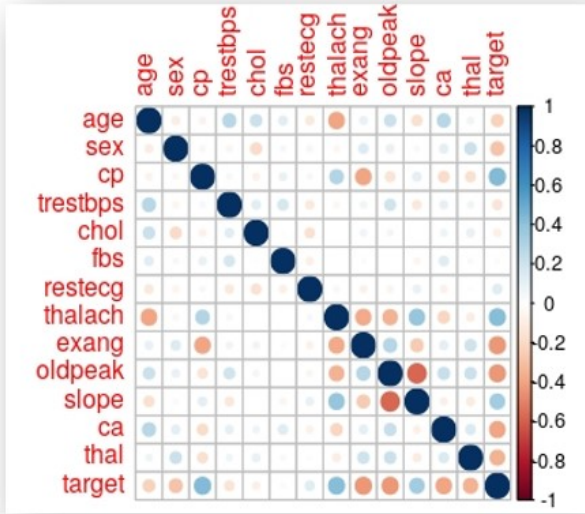


Figure 7: Correlation Matrix

According to the Figure 7 the best predictor variable that can be seen are **cp**, **thalch**, **exang**, **oldpeak**. With "cp" and "thalach" giving positive correlations, while "exang" and "oldpeak" giving negative correlations. For creating the correlation matrix in R **corrplot** package was used.

3.2.2 K-fold cross validation

5 fold cross validation was performed with the mean train **rmse** 0.3708088 and mean test **rmse** of 0.3825556.

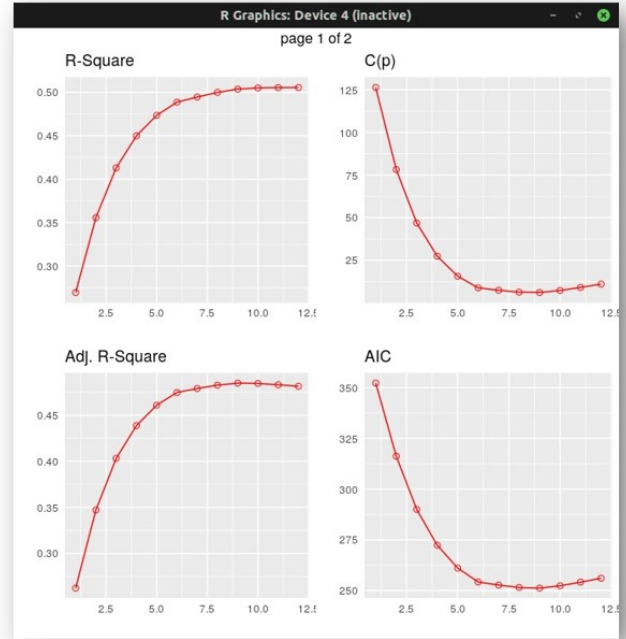


Figure 8: Stepwise model section

3.2.3 Stepwise model selection

From Figure 8, it can be seen from the graphs that 3-4 variables are good for our model as they have low risk of over-fitting and less possibility of error. This stepwise model selection was automated by **ols** [5] package in R which automates the process of selection of predictor variables and runs various permutation and combinations to select the best possible order of variables and generates detailed graphs in various formats showing different kinds of errors.

3.2.4 Stepwise model selection

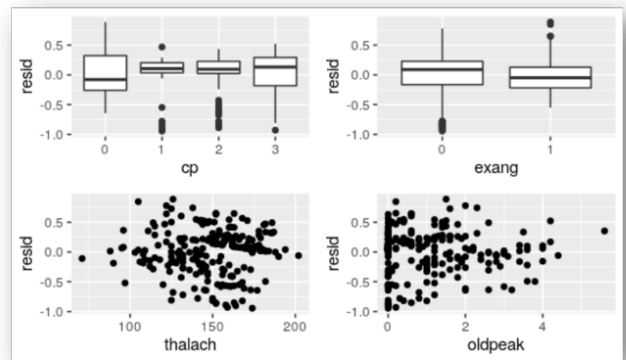


Figure 9: Residuals from linear model

Residuals were also generated for the selected variables over a fit on linear model to further verify the selection of right predictor variables as seen from Figure 9.

3.3 Modelling

3.3.1 Linear Model

```
Confusion Matrix and Statistics

      Reference
Prediction 0 1
0 17  5
1 12 27

      Accuracy : 0.7213
      95% CI : (0.5917, 0.8285)
      No Information Rate : 0.5246
      P-Value [Acc > NIR] : 0.001355

      Kappa : 0.4349

      Mcnemar's Test P-Value : 0.145610

      Sensitivity : 0.5862
      Specificity : 0.8438
      Pos Pred Value : 0.7727
      Neg Pred Value : 0.6923
      Prevalence : 0.4754
      Detection Rate : 0.2787
      Detection Prevalence : 0.3607
      Balanced Accuracy : 0.7150

      'Positive' Class : 0
```

Figure 10: Confusion Matrix: Linear Model

As seen from Figure 10, with linear regression the best model achieved the accuracy of 72.13%. The positive likelihood ratio for this model was 3.7528.

3.3.2 LSSVM

```
Confusion Matrix and Statistics

      Reference
Prediction 0 1
0 21  5
1  8 27

      Accuracy : 0.7869
      95% CI : (0.6632, 0.8814)
      No Information Rate : 0.5246
      P-Value [Acc > NIR] : 2.064e-05

      Kappa : 0.5707

      Mcnemar's Test P-Value : 0.5791

      Sensitivity : 0.7241
      Specificity : 0.8438
      Pos Pred Value : 0.8077
      Neg Pred Value : 0.7714
      Prevalence : 0.4754
      Detection Rate : 0.3443
      Detection Prevalence : 0.4262
      Balanced Accuracy : 0.7839

      'Positive' Class : 0
```

Figure 11: Confusion Matrix: LSSVM

From Figure 11 we can see that LS-SVM achieved an accuracy of 78.69%. Also, the positive likelihood ratio for this model is 4.63572, which is better compared to Linear Model.

3.3.3 Random Forest

```
Confusion Matrix and Statistics

      Reference
Prediction 0 1
0 21  7
1  6 27

      Accuracy : 0.7869
      95% CI : (0.6632, 0.8814)
      No Information Rate : 0.5574
      P-Value [Acc > NIR] : 0.0001579

      Kappa : 0.5697

      Mcnemar's Test P-Value : 1.0000000

      Sensitivity : 0.7778
      Specificity : 0.7941
      Pos Pred Value : 0.7500
      Neg Pred Value : 0.8182
      Prevalence : 0.4426
      Detection Rate : 0.3443
      Detection Prevalence : 0.4590
      Balanced Accuracy : 0.7859

      'Positive' Class : 0
```

Figure 12: Confusion Matrix: Random Forest

As we can see in Figure 12, with Random Forest the accuracy obtained is 78.69%. Surprisingly it has very similar positive likelihood ratio ($Sensitivity/1-Specificity$) for this model is 4.63572.

3.3.4 XGBoost

```
Confusion Matrix and Statistics-----

      Reference
Prediction 0 1
0 20  9
1  8 24

      Accuracy : 0.7213
      95% CI : (0.5917, 0.8285)
      No Information Rate : 0.541
      P-Value [Acc > NIR] : 0.003014

      Kappa : 0.4404

      Mcnemar's Test P-Value : 1.000000

      Sensitivity : 0.7143
      Specificity : 0.7273
      Pos Pred Value : 0.6897
      Neg Pred Value : 0.7500
      Prevalence : 0.4590
      Detection Rate : 0.3279
      Detection Prevalence : 0.4754
      Balanced Accuracy : 0.7208

      'Positive' Class : 0
```

Figure 13: Confusion Matrix: XGBoost

It can be noted from Figure 13 that for XGBoost accuracy obtained XGBoost is 72.13%. The positive likelihood ratio ($Sensitivity/1 - Specificity$) for this model is 2.6193, which lowest of all.

4 Discussion

The models are compared based on 3 parameters: Accuracy, P-value and the positive likelihood ratio ($Sensitivity/1 - Specificity$). Some models have very good accuracy however have a bad positive likelihood ratio, and hence cannot be considered as good model. The predictions from **LS-SVM** is more likely to detect the heart disease given the parameters than any other model.

	LM	LS-SVM	Random Forest	XGBoost
Accuracy	72.13	78.69	78.69	72.13
P-value	0.001355	2.064e-05	0.0001579	0.003014
Likelihood ratio	3.7528	4.63572	3.7775	2.6193

Figure 14: Comparison of various Models

We can conclude that, heart disease is one of the most prevalent diseases and after doing modelling on specifically heart related dataset, it is found that the most contributing factors for heart diseases is chest pressure, maximum heart graph, exercise induced angina and ST depression. It is also noted that medical data modelling can contain high error rates and model accuracies are usually on the lower side.

5 Statement of contributions

It was a joint contribution by all the team members. Everyone was involved in idea development, EDA, education around models, and general discussion of approach.


- **Prakhar Patidar** did Data Pre-processing and Exploratory Data Analysis.
- **Rishabh Shanbhag** worked on Data wrangling and Data visualizations.
- **Nikson Panigrahi** and **Kunjan Khatri** carried out Dimensionality Reduction, Computation of Correlation statistics along with implementation of Linear model, LS-SVM, Random forest and XGBoost.

Equal contribution and efforts by all team members in preparing the presentation and final report.

References

- [1] *Big Cities Health Coalition (BCHC)* 2010-2016, <https://www.bigcitieshealth.org/>.
- [2] *Heart Disease UCI Kaggle*, <https://www.kaggle.com/ronitf/heart-disease-uci>
- [3] *Machine Learning with XGBoost Kaggle*, <https://www.kaggle.com/rtatman/machine-learning-with-xgboost-in-r>
- [4] *"Random Forest in R Towards Data Science*, <https://towardsdatascience.com/random-forest-in-r-f66adf80ec9>
- [5] *Tools for Building OLS Regression Models Cran.r project*, <https://cran.r-project.org/web/packages/corrplot/vignettes/corrplot-intro.html>
- [6] *Spatial Visualization with ggplot2 Cran.r project*, <https://cran.r-project.org/web/packages/ggmap/index.html>

Appendix

 Github: <https://github.com/niksnikson/idmp-bchi>

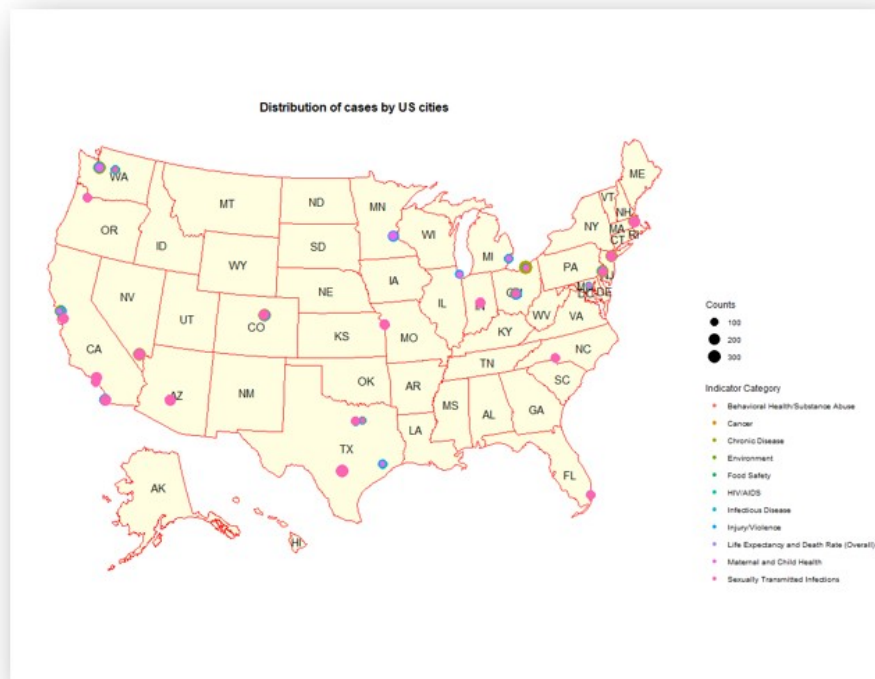


Figure 15: Citywise Distribution



Figure 16: Number of cases for each indicator