# CMPE 255 – Data Mining

# Spring 2018

# School's Progress Prediction Using Data Mining Techniques

By

Sowmya Viswanathan (011432668)

Ishwarya Varadarajan (011549473)

Anupama Upadhyayula (011325041)

# Table of Contents

# 1. Introduction

There are many factors that affects a school's growth rate including locality, population, students' attendance rate, teachers' attendance rate, student and teacher relationship and crime rate. Crime rate of an area is one of the major factors that influences a school's growth rate. Also, students' involvement in crimes can have a direct impact on the performance of a school. Nowadays, in order to prevent students from being involved in crimes, schools have adopted certain rules like compulsory attendance or extra credit for attendance. These can be of great help in improving a school's progress. The location of the school will also be a matter of concern. A school with an unsafe neighborhood, might result in students not attending school regularly, leading to poor grades. This will result in lower performance of the school.

Taking into consideration all these factors, we have decided to predict a school's progress based on the crime incidents that have occurred in its neighborhood. We have considered Chicago's Public-school dataset and crime dataset to find a relation between them. We will be using machine learning algorithms to analyze and predict the progress as 'ABOVE AVERAGE', 'AVERAGE' & 'BELOW AVERAGE'.

# 2. Data

The crime dataset from Chicago's crime data portal - 'https://data.cityofchicago.org/Public-Safety/Crimes-2017/d62x-nvdr' was used. The dataset we obtained contains a lot of missing values and zero values. It is highly imbalanced and irregular. We have also considered the school dataset from Chicago's public-school data portal – 'https://catalog.data.gov/dataset/cps-elementary-school-attendance-boundaries-sy1516'. The school dataset has a very high number of attributes.

All necessary attributes for the prediction were considered for the prediction. There are datasets for crimes from the year 2001 through 2018 and the crime data for each year contains 500k records. Public school datasets of Chicago city data set for the years 2015 – 2017 were considered. The distances were computed for each school with each row of the crime dataset and concatenated the rows based on that.

**Chicago Crime data set**

**URL :** https://data.cityofchicago.org/Public-Safety/Crimes-2001-to-present/ijzp-q8t2/data

**Chicago Public Schools - School Progress Reports SY1516 & SY1617**

https://data.cityofchicago.org/Education/Chicago-Public-Schools-School-Progress-Reports-SY1/fvrx-esxp

https://data.cityofchicago.org/Education/ChicLibraries go-Public-Schools-School-Progress-Reports-SY1/cp7s-7gxg

# 3. Requirements

Python was used as the primary language for the execution of the algorithm. Since, the data sets are having high number of records, we transferred our files from local to the HPC server provided by the University and executed our code on the HPC server – Linux based O.S. To display the result of our predicted values, we have written a simple HTML web page that is built using Flask framework to run the algorithm and displays predicted output values.

## 3.1. Libraries:

**Pandas**: We have imported our files as CSV files, read them into a pandas data frame and transferred the CSV files by making use of data frames. Hence, we extensively used pandas in our algorithms.

**Numpy**: In order to increase the computational speed and to interface with different packages, we used Numpy in our algorithms.

**Scipy**: Our data contains a lot of values with different ranges. To set them on the same scale and avoid much differences in calculations, we have imported Normalizer class from scipy learn.

**Xgboost**: We used three classifiers to train and predict the values of a given dataset. xgboost has been considered as one of our classifiers to train the dataset – independent attributes and test the values of dependent variables. After predicting the values, we used classifier to test the accuracy. We installed XGBoost classifier on the HPC and then ran the algorithm.

**Random forests**: We used random forest as the other classifier to predict the values of a given dataset by training them. Number of trees considered for the prediction was 100 and the criterion 'Entropy' was used as a parameter. We have considered different parameters to tune in the values.

## 3.2. HPC

Our dataset contains 400k records. We optimized the records and handled missing values but the data sets are so huge, we needed a much powerful machine. We used HPC to run the algorithm by transferring the files from local to HPC by using the SCP command. After transferring the files, we have loaded the Numpy module and ran the python algorithm.

## 3.3. Data Preprocessing:

One of the fields of the crime data has Primary_Type attribute which represents the type of crime. We had to create dummy variables for all the values of primary_type attribute using pandas. Onehot encoder and label encoder was attempted to be made use of in the process. But it was not feasible with the data we have. Therefore, we used pandas to get dummy variables.

### 3.4. Web Application:

A HTML web page was created that displays the predicted value of the school that the user enters. When the user hits the submit button, Random Forest Algorithm runs in the background to predict the school's growth rate based on its training with past data. This was developed using Flask framework.

## 4. Workflow of the algorithm

### 4.1. Extracting the datasets

The first step of any data mining algorithm is to extract the data sets and perform the necessary preprocessing steps. The data sets we obtained are two independent datasets which have no relation to each other except for latitudes and longitudes. To define a relationship between the two datasets we used the latitudes and longitudes that are present in both the datasets and we defined a program that calculates the distances and based on the distances we check the neighbors for each school set – different rows of the crime datasets.
The data we obtained was very imbalanced and therefore extracting and performing required operations on the crime datasets and school dataset takes a lot of time.

### 4.2. Preprocessing the data

Data preprocessing is one of the important of any machine learning algorithm. Data without preprocessing is very difficult to handle and moreover it impacts the calculation. The datasets we obtained from the government portal contains a lot of missing values. Therefore, the first step in the preprocessing step was to handle the missing values. All the rows are independent of each other therefore we replaced the missing values in the data by '0'. Replacing with mean, standard deviation gave higher error rate.

After replacing the missing values, we created dummy variables for a column 'Primary_Type'. Primary type has 34 different fields. Therefore, we used pandas to create dummy variables for each field and concatenated the dummy variables with the original file. We have taken the dependent attribute of the school data set -School's performance and named it as y_train. Y_train is the attribute for which predictions must be performed.

Some of the attributes of the dataset have different ranges. Therefore, to have them on the same range we normalized the independent variables with different ranges and concatenated them to the previous dataset.

# 5. Data Classification

## 5.1. Random Forest Classifier

An ensemble method of classification, which selects subsets of features and builds decision trees.
We considered and modified the following parameters of sklearn's RandomForestClassifier:
n_estimators
criterion
max_depth

## 5.2. XGBoost Classifier

Extreme Gradient Boosting method is a supervised learning model. We first installed the classifier on HPC system using pip install and once installed we imported the module to our algorithm and ran it on the dataset. The following parameters of XGBClassifier() function were considered and modified for improving the prediction accuracy:
learning_rate
n_estimators
max_depth

## 5.3. Naïve Bayes

It is a type of supervised machine learning algorithm. We used sklearn's linear model to import the classes and considered the following types of Naïve Bayes classifier:
GaussianNB

## 5.4. SDG Classifier

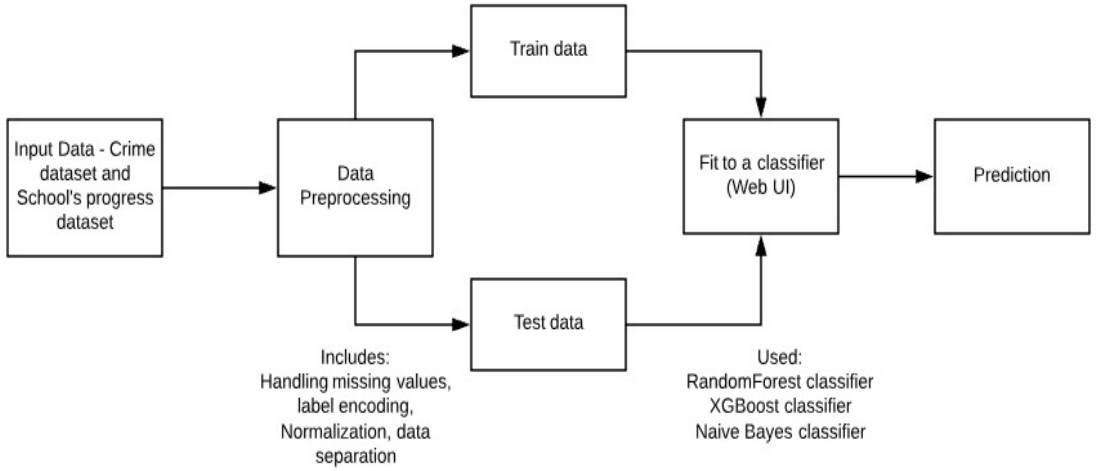
It is a type of linear model for classification. Implements the algorithm using the Stochastic Gradient Descent learning. The parameters using for tuning the algorithm are:

- Loss
- Penalty
- Alpha
- max_iter

The different parameters were modified in multiple iterations to tune our algorithm to provide a better output which is more accurate.

# 6. Project Architecture

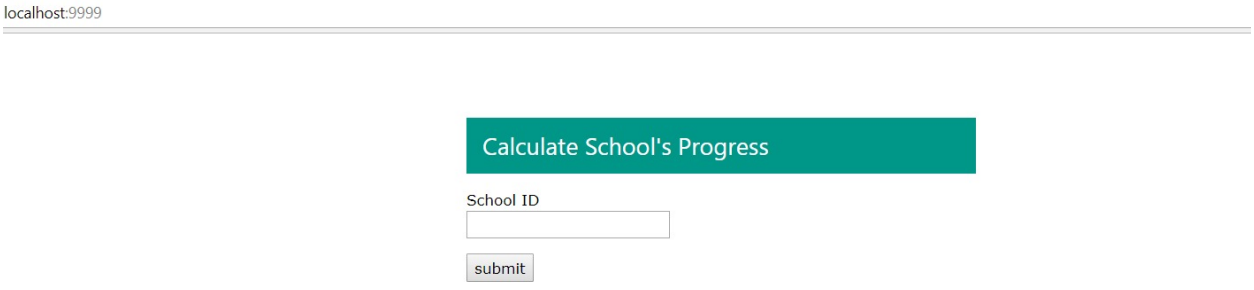
The below shown figure shows the architecture of our workflow. The data that we get from the government portal will be pre-processed, fit into a classifier and the results will be predicted. The web UI contains a button which upon clicking runs the classifier on the data and displays the prediction.

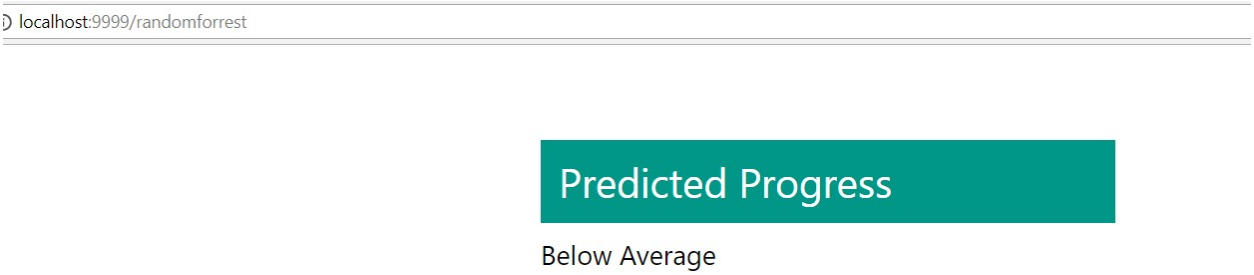



The dataset we obtained contains a lot of values and the processing time is more. Initially the threshold for distances was 11 and the size of files generated for that was so high around 2GB. herefore, we reduced the size of the dataset to be 3 and preprocessed the data based on this threshold. Predictions were considered based on this threshold.

## 7. Web UI Design

The UI was designed using HTML, CSS, Flask framework, and python for scripting. Below is a very simple representation of the UI.






# 8. Results

We are using different classifiers to get the predictions of school's performance. The following

## 8.1. Random Forests Classifier:

We verified for a couple of School IDs, if the actual and expected progress were correct and found that it produced the expected results.

But the accuracy when calculated was found to be lower than 50%.

| | |
|---|---|
| Accuracy: 0.47385 | Parameters: n_estimators = 5, criterion = 'entropy' |
| Accuracy: 0.4707 | Parameters: n_estimators = 10, criterion = 'entropy' |
| Accuracy: 0.4777 | Parameters: n_estimators = 200, criterion = 'entropy' |

## 8.2. XGBoost Classifier:

| | |
|---|---|
| Accuracy: 0.4152 | Parameters: n_estimators=50, learning_rate=0.3, max_depth=8, min_child_weight=2, objective= 'binary:logistic' |

| | |
|---|---|
| Accuracy: 0.3267 | Parameters: n_estimators=10, learning_rate=0.3, max_depth=8, min_child_weight=2, objective= 'binary:logistic' |

From the above results, the accuracy value that has been obtained from the prediction is shown to be good using the 'Random Forests' classifier. Hence, the prediction for the school's performance is giving better results using the Random Forest classifier taking crime data as the base.

# 9. Challenges faced and dependencies

## 9.1. Challenges faced:

As the data was huge, we faced a lot of problems in the pre-processing steps. A summary of the challenges is mentioned below:
The calculation of the distance between location of a crime incident and a school, tool a lot of time, almost close to 8 hours.

Many steps were taken to reduce the number of hours it took to execute the code. Some of which are:

- Optimized the code, to reduce the number of steps it took to write to an output file.
- We divided the data set into halves and ran it on different HPC instances, which considerable reduced the total hours to half.
- Initially the threshold for distances was taken as 11 and it added almost 5k records for each distance and hence we reduced it to 3. This reduced the computation.
- The test data, train data obtained from each of the 2015/2016 data sets of both crime and school's progress had each over 1 million rows of data. Hence, we decided to use half data of 2015-16 to predict for 2016-17 data.

As the data was huge, even after deciding to use half of the actual data for test and as train data, the classifier algorithms took a lot of time to produce results.
RandomForest Classifier for 200 estimators- it took approximately 15 mins to run.
XGB classifier too more than 15min to fit the train and test data sets and predict the values for test datasets.

# 10. Dependencies:

The size of the datasets were huge and complex computations were performed on them. Hence, computation required high performance machine. We were highly dependent on the high-performance computing system. We used the HPC server provide by the professor and ran

the datasets taking 5 Nodes. We followed the instructions provided by the professor to compute the results.

## 11. Conclusion

Most of the School IDs which were used to find the actual progress rate, produced the expected results. This occurred while using the Randomforest classifier. SGD classifier's accuracy of prediction as low as for most of the records it gave 'AVERAGE' as the progress of the school which was incorrect prediction.

On calculating the accuracy values for different classifiers, it was found to be comparatively lower than expected.

The crime incidents that happen around a particular neighborhood does seem to have a relation to the school's progress. We would require more school data and crime data, to train the models, in order to get better results.

## 12. References:


- https://eml.berkeley.edu/~moretti/lm46.pdf
- https://www.tandfonline.com/doi/abs/10.1080/00036840701604412?journalCode=raec20
- http://www.nber.org/papers/w8605