CS 760: Machine Learning Fall 2020

Predicting female literacy rate and school dropout rate in India

Authors: Bhagyashree Prabhune (bprabhune@wisc.edu) and Saketh Sridhara (ssridhara@wisc.edu)

Abstract:The motivation for this project was to understand one of the key reasons behind the low literacy rate(i.e., high dropout rates amongst schoolchildren) in India. This project aims to predict the dropout rates and enrollment rates among schoolchildren in India, taking into account the large amount of state-wise and district (county) level data collected over a variety of parameters of the current Indian education system. First off, we work with state-wise data over three years of the Indian education system. We identify key features pertaining to the schools, and predict the dropout rates from these features through the linear regression technique. It is seen that the most significant feature is the enrollment rate i.e, higher the enrollment in a state, lower is the dropout rate. For any given feature vector input, the confidence interval for the dropout rate is also established. Next, having identified enrollment rate as the most important factor for increasing the literacy rate (i.e. reducing dropouts), we develop a linear regression model which predict enrollment rates from various parameters at the district level (using a larger dataset). We find that the most significant feature affecting the enrollment in schools as the female literacy rate of the district. Further, the female literacy rates were predicted using SVM, random forest and neural networks. All these ML techniques are in agreement and predicted similar female literacy rate for a feature vector. Also, the significant features affecting female literacy rates are availability of electricity and number of schools with only one teacher. By providing access to electricity and converting the schools from single to multiple teacher schools can help bolster the primary education system of India. This helps states and districts focus on improving the identified resources/infrastructure, which would result in driving up literacy and reducing dropouts. The enrollment and dropout predicting programs can also be utilized to set targets for each district/state in the country.

1 Introduction

India is a country of 1.3 billion people, where 67% of the Indian population lives in rural areas. Approx. 196 million elementary school going children, out of that 146 million are enrolled in rural schools (Source: U-DISE 15-16). The school education in rural India is mostly dependent on Government and Government aided schools. For rural India journey of education is not easy, children from rural areas face many challenges till they finish their education. The top challenges faced by schoolchildren are -

1. Lack of electricity and drinking water in schools
2. Lack of proper sanitation facilities (for boys and girls)
3. Lack of an access to computers and advanced learning tools (digital literacy tools)
4. Lack of good infrastructure such as roads and playgrounds
5. Lack of teachers and classroom facilities

And more. These challenges deter children from completing their schooling, and hence they dropout, or worst case, don't enroll to schools. Due to dropouts and reduced enrollments, India's literacy rate stands at about 74%. Lower literacy can have detrimental effects on the society, such as increased crime rates, unemployment etc. Hence, it is of utmost importance for a country like India to drive up the enrollment to schools and reduce the droputs. The project aims at using Machine Learning methods to predict dropout rates and enrollment rates based on datasets collected at the state-level and district level across India. The features of the dataset correspond closely the to challenges listed above, and the response we predict are the dropout rates and enrollment rates.

Link to the news article from where a portion of this introduction has been adopted.

2 Related/Similar work

Dropout rates have been predicted using machine learning techniques, albeit in a different setting, typically for online learning (where dropout rates were found to be significantly high). Kloft et.al [1] predicted the dropout rates in MOOC over weeks using Support Vector Machines. Kotsiantis et al. [2] predicted preventing student dropouts in distance learning using Naive Bayes algorithm. Tan et al. [3] studied high dropout rates in E-learning programs, and developed prediction models using Neural Networks, Decision Trees and Bayesian Networks with a large sample of over 60,000 students, and concluded while all the three methods were effective in prediction, decision tress gave the best performance. Lykourentzou et. al. [4] using a combination of SVM, Neural Nets and a probabilistic ensemble simplified fuzzy ARTMAP to predict dropout rates for e-learning courses. Fetler et. al. [5], correlated dropout rates, academic performance and poverty using various study variables in the late 80's. The biggest takeaway from literature review was that most ML methods were doing a reasonable job in prediction of dropout rates. We intend to apply a couple of ML techniques and report their accuracy in predicting dropout rates and enrollment rates among schoolchildren in India.

We are demonstrating variety of ML methods on the data set containing primary education related information from India to make inferences which can aid in it's improvement.

3 Dataset

The aim of this project was to get insights into the dropout and enrollment rates among schoolchildren in India. Hence, we have picked up two relevant datasets, one of which was useful in predicting dropout rates and the other in predicting enrollment rates.

We obtained both the datasets through Kaggle, which have obtained information from the Govt. of India's website https://data.gov.in/.

The smaller dataset State-level dataset consists of three years of school performance data across the states, and has a total of about 110 data-entries. For the purpose of predicting the dropout rates, a random category of students was chosen (in this case, Secondary Boys), and the entries corresponding to this category was extracted. Missing data was populated with average school data across all categories. All the data is in percentage, hence it is in the same scale. The corresponding CSV file can be found here.

The larger dataset District-level dataset contains over school information data from over 600 districts in India. This was used for the analysis on enrollment rates and Female literacy. The corresponding CSV file can be found here

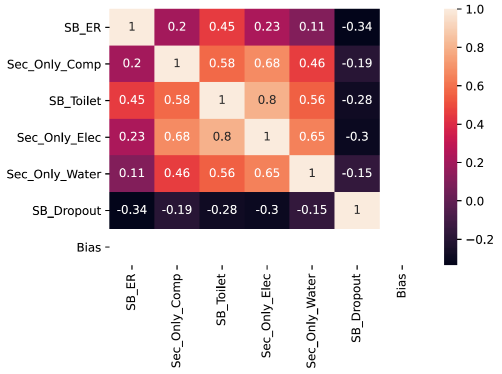
4 Approach

The smaller state-level dataset was used to predict the dropout rates, and the larger district-level dataset was used to predict the enrollment rates. With the smaller dataset, dropout rates was predicted using the linear regression model. With the larger dataset, enrollment rates was predicted, using linear regression model. By analyzing the results obtained through linear regression, we found that female literacy rate is the most significant feature affecting the enrollment rates. Hence the female literacy rate was predicted using support vector machine (SVM), random forest (RF) and neural networks (NN).

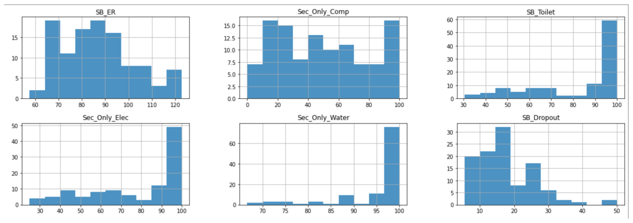
4.1 Predicting dropout rates using linear regression

The features in the state-level dataset are: The enrollment rates and percentage of schools with computers, toilets , electricity and drinking water supply The response was chosen to be the dropout ratio.

First, a heatmap was created to visualize the correlation between the features and the response. The histogram was also plotted for the dataset. It is shown inFigure 1.



(a) Correlation Heatmap



(b) Histogram

Figure 1: State data

It is clear from the correlation heatmap that the dropout rates (last row) is dependent on all the chosen features substantially and is negatively correlated. This is in accordance with the intuition, i.e., if there is higher enrollment, higher number of schools with computers, electricity, toiltes, and drinking water, lower is the dropout rates in that state. Additionally, there appears no high-correlation ($\geq$0.7) amongst the chosen features.

We used the linear regression codes developed in class for prediction of dropout rates, identification of significant features and providing a confidence interval for the prediction. The results were compared with the outputs from scikit-learn [6] using its LinearRegression function for verification purposes.

4.2 Predicting enrollment rates using linear regression

The same procedure as mentioned in Section 4.1 is repeated with the larger district-level dataset. The features in the district-level dataset are:

- Schools with boy's toilets
- Schools with playground
- Schools with drinking water
- Schools with Roads
- Classrooms requiring major repair
- Schools with Electricity
- Schools with a single teacher
- Female literacy rate

The response was chosen to be the enrollment rate, and the correlation heatmap (Figure 2 was studied similarly.

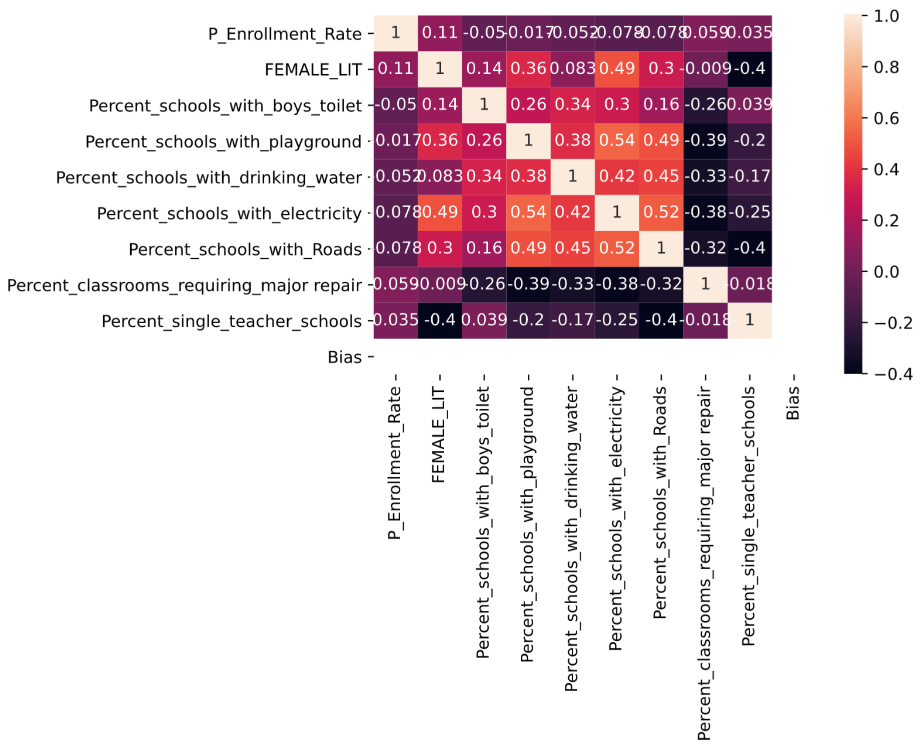


Figure 2: Correlation heatmap from district data

Some of the correlations made sense, like the enrollment rate was positively correlated with female literacy rate, and negatively correlated with respect to infrastructure related features. However, there were no other significant insights from the chosen features. The linear regression implementation and verification follows the same as Section 4.1.

4.3 Predicting female literacy rates using Support Vector Machines

From the correlation matrix shown in Fig. 2, female literacy rate has a high correlation with many features like percentage of schools with single teachers, percent schools with all-weather roads, percent schools with playground, and percent school with electricity. So we find the effect of other features on the female literacy rate of a district (county). In other words, we predict the female literacy rate as a function of following 6 parameters:

- Schools with boy's toilets
- Schools with playground
- Schools with drinking water
- Schools with Electricity
- Schools with all-weather roads
- Schools with a single teacher

SVM is used as regressor to predict the female literacy rate using Scikit-learn library. K-fold cross validation was conducted.

- **Data preprocessing**: The input data was scaled using min-max scaler before feeding to the SVM regressor. Principal component analysis was performed on the input. It showed that all the current features were crucial However, it was found that the number of inputs were already optimum. So none of the features were eliminated.
- **Metric used to evaluate the performance**: It is important to know the correlation between the

predicted and actual output. Hence Pearson correlation coefficient (PCC) is better suited and is used in this study to evaluate the performance of the network. It measures the linear correlation between the predicted and true target variable. PCC has values between -1 and +1. +1 indicates positive linear correlation while -1 indicates that variables are negatively correlated. 0 value of PCC indicates the variables are not correlated.

- **Kernel**: Linear as well as non-linear kernel can be used. Here we used the non-linear 'radial basis function' kernel as it is better suited for this data.
- **Best hyperparameters for RBF SVM**: The performance of RBF SVR depends on the parameters like 'gamma' and 'C' values. The behavior of the model is sensitive to the gamma parameter. Gamma determines extent of the influence of a single training example. Also, smaller C provides higher bias and lower variance and vice versa. Best parameters for the model were found by the 'Grid search' function provided by Sklearn. For a given set of gamma and C values, grid search finds the combination which provides the highest score. For this model, the best parameters were: 'C': 10, 'gamma': 5, with the score of 0.37.

4.4 Predicting female literacy rates using Random Forest

In order to verify the prediction of SVM, we employ random forest (RF) and NN. RF fits a number of decision trees on the sub-samples of the data to control the over-fitting while improving the accuracy. Data-preprocessing and performance evaluation matrix as mentioned in SVM regressor were used for the random forest model. RF with the following parameters was implemented using 'SKLearn' library. Grid search was employed to find the best parameters:

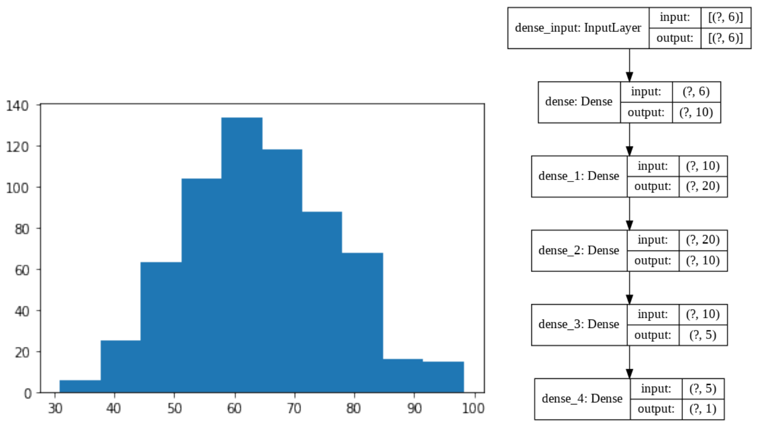
- **Number of trees in the forest**: The parameter 'n_estimators' provides this information. Using grid-search, the number of trees selected were 1000.
- **Quality of split**: The default parameter 'mean squared error' was used to measure the quality of the split.
- Minimum number of samples required to split an internal node: The variable 'min_samples_split' governs this parameter and was set to 3 by the grid-search.
- **Maximum depth**: The maximum depth chosen by the Grid-search was 'None' which indicated that the RF gave maximum accuracy when the nodes were expanded until all leaves of decision trees are pure or until all leaves contain less than 'min_samples_split' samples.

4.5 Predicting female literacy rates using Neural Networks

Data-preprocessing and performance evaluation matrix as mentioned in SVM regressor were used for this neural network (NN) model. NN with following configuration was employed using 'keras' library:

- **NN architecture**: In this NN architecture (see Fig. 3a), we gradually increase and then decrease the embedding length of the network so that every layer gains meaningful information from the previous layer and produces relevant embedding. With the input dimension = 6, the first layer has 10 nodes, followed by20-10-5-1 nodes for subsequent layers. The output layer has one node for the target variable.

- **Activation function**: Here, we are predicting a continuous numerical value which lies in the range of 0 to 100. 'Relu' activation function was used for all the layers.
- **Optimizer**: Adams optimizer was used with an adaptive learning rate. It uses a stochastic gradient descent algorithm to minimize the loss function and is based on adaptive estimation of first and second order moments. Adaptive learning rate was employed by as following. The learning rate for the first 2000 epochs was mentained at 10^{-3}, and then decreased to 10^{-4} for the next 500 epochs.



(a) Distribution of female literacy rate

(b) Neural network architecture

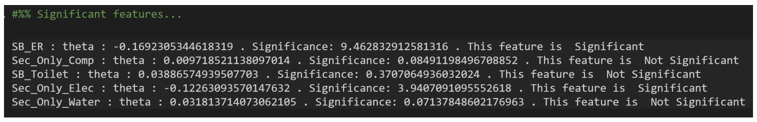
Figure 3: State data

- **Loss function**: Popular regression loss functions include: Mean squared error, Mean absolute error, Mean squared logarithmic error. Among these, the mean squared logarithmic error is suitable if the data has a large spread of values. Values in the present data have a limited spread – all the values lie between 0 to 100. Moreover, the distribution of the target variable is more or less Gaussian as shown in Fig. 3b, Hence, mean squared error loss function was employed to predict the female literacy rate.

5 Results

5.1 Prediction of dropout rate(states) and enrollment rate(districts)

The dropout rate prediction with linear regression is as follows. For feature vector: $\mathbf{x} = \begin{bmatrix}100 & 20 & 90 & 90 & 90\end{bmatrix}$ The output (dropout prediction) using linear regression are: Predicted dropout rate: 14.105 with confidence interval ($\alpha = 0.05$): 10.476% - 17.734%. Looking at feature significance:

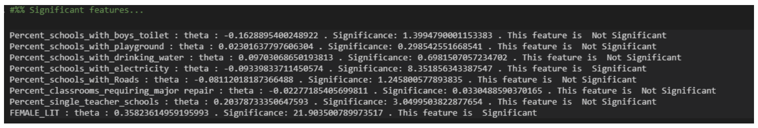


```
#%% Significant features...

SB_ER : theta : -0.1692305344618319 . Significance: 9.462832912581316 . This feature is  Significant
Sec_Only_Comp : theta : 0.009718521138097014 . Significance: 0.08491198496708852 . This feature is  Not Significant
SB_Toilet : theta : 0.03886574939507703 . Significance: 0.3707064936032024 . This feature is  Not Significant
Sec_Only_Elec : theta : -0.12263093570147632 . Significance: 3.9407091095552618 . This feature is  Significant
Sec_Only_Water : theta : 0.031813714073062105 . Significance: 0.07137848602176963 . This feature is  Not Significant
```

Figure 4: Feature significance for dropout prediction

The results and feature significance show that the **most significant factors influencing dropout rates are the enrollment rates and availability of electricity in schools.** To reduce dropout rates amongst schoolchildren in India, we will need to increase the enrollment rates and increase the number of schools with electricity.
Prediction of enrollment rates (districts): Results from linear regression show that the most significant feature influencing enrollment rate was female literacy rate and the percentage of schools with electricity. The other features showed lower significance. However, the R^2 values of the fit was not high, close to zero. And a cross-validation score ($\leq 40\%$) while predicting within the confidence interval. Thus, linear regression was not the most suitable tool in predicting enrollment rates with these features.



```
#%% Significant features...

Percent_schools_with_boys_toilet : theta : -0.1628895400248922 . Significance: 1.3994790001153383 . This feature is  Not Significant
Percent_schools_with_playground : theta : 0.02301637797606304 . Significance: 0.298542551668541 . This feature is  Not Significant
Percent_schools_with_drinking_water : theta : 0.09703068650193813 . Significance: 0.6981507057234702 . This feature is  Not Significant
Percent_schools_with_electricity : theta : -0.09339833711450574 . Significance: 8.351856343387547 . This feature is  Significant
Percent_schools_with_Roads : theta : -0.08112018187366488 . Significance: 1.245800577893835 . This feature is  Not Significant
Percent_classrooms_requiring_major repair : theta : -0.02277185405699811 . Significance: 0.0330488590370165 . This feature is  Not Significant
Percent_single_teacher_schools : theta : 0.20378733350647593 . Significance: 3.0499503822877654 . This feature is  Not Significant
FEMALE_LIT : theta : 0.35823614959195993 . Significance: 21.903500789973517 . This feature is  Significant
```

Figure 5: Feature significance for enrollment prediction

5.2 Prediction of female literacy rate

5.2.1 With SVM

The model was tested and 5-fold cross validation was performed by splitting the data into training and testing sets with the ratio 80:20. Pearson correlation coefficient (PCC) for the predicted and test data of the target variable 'female literacy rate' was obtained as

$$\mathrm{PCC_{svm}} = 0.5185$$

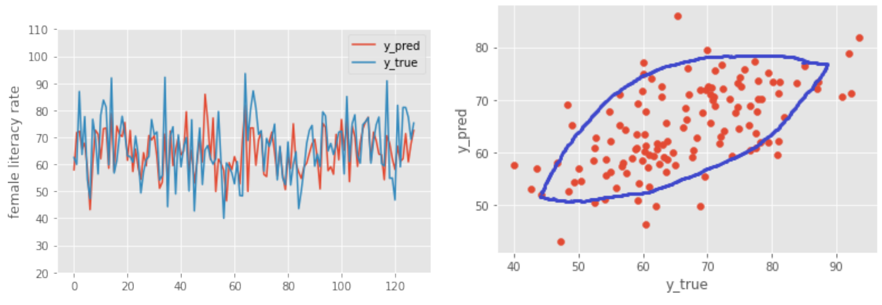
For a feature vector: $\mathbf{x} = \begin{bmatrix}100 & 20 & 90 & 90 & 90\end{bmatrix}$, the predicted female literacy rate by SVM = 63.55

5.2.2 With Random Forest

Random forest proves to be a better model compared to SVM for this data providing PCC value of

$$\mathrm{PCC_{RF}} = 0.5936$$

The plot of predicted and actual values of female literacy rate are shown in the Fig. 6. The values show a good correlation as observed from PCC value as well as the plots.



(a) Predicted and actual values (b) Scatter plot

Figure 6: Plots of predicted and actual values of target variable for RF

The blue colored ellipse like outline drawn in Fig. 6b depicts the correlation between the predicted and true values of the target variable. If the outline is narrow and inclined at an angle close to 45^0, it indicates that the two variables are perfectly correlated. If the outline tends towards a circle, it indicates that the variables are not correlated. Similarly, if is the region is narrow and inclined towards negative 45^0, it indicates that the variables are negatively correlated. From Fig. 6b, we can see that the variables are positively correlated as expected with the inclination close to 45^0.

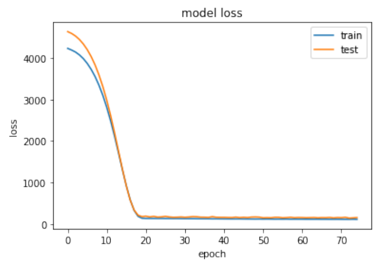
For the same feature vector, the predicted female literacy rate by RF = 67.43

5.2.3 With Neural Network

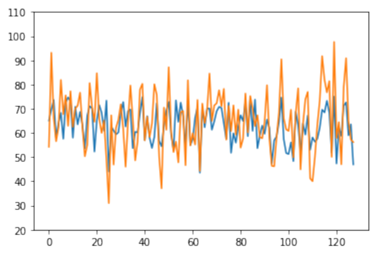
Next, the NN was implemented for predicting the female literacy rate. The decrease in loss function for various epochs is shown in Fig. 7a. Only 75 epochs are shown in the figure. Actual model was run for more than 2500 epochs. We obtain a higher PCC value from NN compared to RF and SVM:

$$\text{PCC}_{\text{NN}} = \mathbf{0.6849}$$

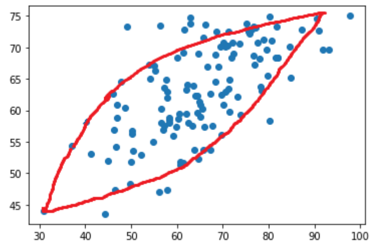
The scatter plot of predicted and true values of the target variable is shown in Fig. 7c. The red outline (similar to the blue outline in Fig. 6c) indicates the correlation between the two vectors. Compared to Fig. 6c, the outline in the Fig. 7c is narrower which indicates a better correlation between the two vectors. This is reflected in the better PCC value obtained by using NN.



(a) Loss Vs Epoch



(b) Predicted and actual values



(c) Predicted and actual values

Figure 7: Neural network model results

R^2 value obtained for NN model = **0.44**. For the same feature vector, the predicted female literacy rate by NN = 69.40 which is closer to that obtained using RF.

6 Conclusion and Future Work

Through this project, we have analyzed few of the factors influencing the dropout rates and enrollment rates amongst the primary schoolchildren in India. Through a linear regression model on the state-wise data, it is seen that dropout rates depend on the enrollment rates and the availability of electricity in schools. It is necessary for the government to increase awareness in enrolling more children to schools and providing basic infrastructure such as electricity. Higher enrollment rates show lower dropouts, thus we attempted to identify what factors affect the enrollment rates amongst children with a district level. We found that enrollment rate is heavily dependent on the female literacy rate of the district and very weakly dependent on the other factors. This supports the fact that female literacy is crucial for encouraging more children to complete their schooling. We further analyzed the factors affecting the female literacy rate in the district. The factors like availability of electricity, clean drinking water, playgrounds, sanitation (availability of toilets) are significant in predicting the female literacy rate. Neural network provided the best accuracy and R^2 value followed by random forest and support vector machines. The most significant feature affecting the female literacy is availability of electricity in schools. The second most significant feature is the schools with only one teacher. Many primary schools in India have only one teacher to teach all the subjects for all the classes. Such a ardous work naturally causes strain on the teacher and effects efficiency. If the schools employ more than one teacher, it would help in increasing the female literacy rate. Efforts targeted towards female education through, say more number of evening schools targeted for women, increasing awareness of education among women would bolster the educational system of India.

Appendix A Predicting state-wise dropout rates using Linear Regression

Link to the source dataset: Link to Kaggle.

CSV data extracted from source link and used for the experiment: Click to view CSV

Code can be downloaded from: Click to download code

```
#%% ME760 Project - Predicting dropout rates from state-wise education data
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
from sklearn.metrics import mean_squared_error, r2_score
from scipy.stats import norm
from scipy.stats.distributions import chi2

import math

#%%
def read_data(filename):
    df = pd.read_csv(filename)
    df['Bias'] = 1;
    return df

#%%
def linear_regression(my_xvector,df):
    print('Linear regression using own code:')
    y = df['SB_Dropout'].values
    X = df[['Bias','SB_ER','Sec_Only_Comp','SB_Toilet','Sec_Only_Elec','Sec_Only_Water']].values
    theta_hat = np.matmul(np.linalg.inv(np.matmul(X.T,X)),np.matmul(X.T,y));
    N = len(y);
    my_xvector = np.insert(my_xvector, [0], 1);
    y_pred = np.matmul(my_xvector.T, theta_hat);
    print('predicted dropout rate:',y_pred)
    var_hat = (1.0/N)*np.matmul((y - np.matmul(X,theta_hat)).T,(y - np.matmul(X,theta_hat)));
    # print('var_hat',var_hat)
    temp = var_hat*np.matmul(my_xvector.T,np.matmul(np.linalg.inv(np.matmul(X.T,X)),my_xvector));
    tau = norm.ppf(alpha/2,loc = 0,scale = math.sqrt(temp.item(0)))
    # print('tau',tau)
    print('confidence interval:',(y_pred+tau),(y_pred-tau))
    return y_pred
#%%
def LR_significance(my_xvector,df,alpha):
    y = df['SB_Dropout'].values
    Col_names = ['Bias','SB_ER','Sec_Only_Comp','SB_Toilet','Sec_Only_Elec','Sec_Only_Water']
    X = df[Col_names].values

    theta_hat = np.matmul(np.linalg.inv(np.matmul(X.T,X)),np.matmul(X.T,y));
    N = len(y);
    my_xvector = np.insert(my_xvector, [0], 1);
    y_pred = np.matmul(my_xvector.T, theta_hat);
    var_hat = (1.0/N)*np.matmul((y - np.matmul(X,theta_hat)).T,(y - np.matmul(X,theta_hat)));
    # print('var_hat',var_hat)
    temp = var_hat*np.matmul(my_xvector.T,np.matmul(np.linalg.inv(np.matmul(X.T,X)),my_xvector));
    tau = norm.ppf(alpha/2,loc = 0,scale = math.sqrt(temp.item(0)))
    # print('tau',tau)
    # print('confidence interval:',(y_pred+tau),(y_pred-tau))
    #2.7 d and e significant feature height and weight
    cov_theta = var_hat.item(0)*np.linalg.inv(np.matmul(X.T,X));
    # d = 3.841 # from chisquare tables
    cutoff = chi2.ppf(1 - alpha, df=1)

    test_stat_out = np.zeros(len(theta_hat));
    for i in range(1,len(theta_hat)):
        test_stat_out[i] = pow(theta_hat[i]/math.sqrt(cov_theta[i,i]),2)
        if test_stat_out[i] > cutoff:
            is_sig = 'Significant';
        else:
            is_sig = 'Not Significant';
        print(Col_names[i],': theta :',theta_hat[i] ,'. Significance:',test_stat_out[i],'. This
             feature is ',is_sig)
    return var_hat


#%%
def sk_linear_regression(my_xvector,df):
    print('Using Scikit linearmodel regression')
    y = df['SB_Dropout'].values
    X = df[['SB_ER','Sec_Only_Comp','SB_Toilet','Sec_Only_Elec','Sec_Only_Water']].values
    reg = linear_model.LinearRegression();
    reg.fit(X,y)
    y_pred = reg.predict(my_xvector)

    print('Coefficients: \n', reg.coef_)
    print('Intercept: \n', reg.intercept_)


    print('Prediction: \n', y_pred)
    return y_pred
#%% data and histograms
filename = 'school_dataset.csv'
df= read_data(filename)
df.hist(alpha=0.8, figsize=(20, 10))
# ['SB_ER','Sec_Only_Comp','SB_Toilet','Sec_Only_Elec','Sec_Only_Water']
my_xvector = np.array([100,20,90,90,90]).reshape(1, -1);

#%% Predict dropout rate using Scikits Linear regression
# All values in percentage/ratio
y_out = sk_linear_regression(my_xvector,df)
# %% PRedict with Linear regression
y_out = linear_regression(my_xvector,df)
#%% Significant features
alpha = 0.05;
var_hat = LR_significance(my_xvector,df,alpha)
# %%
```

Appendix B Predicting district-wise enrollment rate using Linear Regression

Link to the source dataset: Link to Kaggle.

CSV data extracted from source link and used for the experiment: Click to view CSV

Code can be downloaded from: Click to download code

```
#%% ME760 Project - Predicting enrollment rates from district-wise education data
#%%
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model, metrics
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from scipy.stats import norm
from scipy.stats.distributions import chi2
import seaborn as sn

import math

#%%
def read_data(filename):
    df = pd.read_csv(filename)
    df['Bias'] = 1;
    return df
#%%
def read_data_with_fields(filename,fields):
    df = pd.read_csv(filename,usecols = fields)
    df['Bias'] = 1;
    return df
#%%
def linear_regression(my_xvector,df,alpha,fields):
    # print('Linear regression using own code:')
    y = df[fields[0]].values
    # fields.insert(1,'Bias')
    Col_names = fields[1:len(fields)]
    X = df[Col_names].values
    theta_hat = np.matmul(np.linalg.inv(np.matmul(X.T,X)),np.matmul(X.T,y));
    print(theta_hat)
    N = len(y);
    my_xvector = np.insert(my_xvector, [0], 1);
    y_pred = np.matmul(my_xvector.T, theta_hat);
    # print('predicted enrollment rate:',y_pred)
    var_hat = (1.0/N)*np.matmul((y - np.matmul(X,theta_hat)).T,(y - np.matmul(X,theta_hat)));
    # print('var_hat',var_hat)
    temp = var_hat*np.matmul(my_xvector.T,np.matmul(np.linalg.inv(np.matmul(X.T,X)),my_xvector));
    tau = norm.ppf(alpha/2,loc = 0,scale = math.sqrt(temp.item(0)))
    # print('tau',tau)
    # rint('confidence interval:',(y_pred+tau),(y_pred-tau))
    y_lb = y_pred+tau
    y_ub = y_pred-tau
    return y_pred,y_lb, y_ub
#%%
def LR_significance(my_xvector,df,alpha,fields):
    y = df[fields[0]].values
    # fields.insert(1,'Bias')
    Col_names = fields[1:10]
    X = df[Col_names].values

    theta_hat = np.matmul(np.linalg.inv(np.matmul(X.T,X)),np.matmul(X.T,y));
    N = len(y);
    my_xvector = np.insert(my_xvector, [0], 1);
    y_pred = np.matmul(my_xvector.T, theta_hat);

    var_hat = (1.0/N)*np.matmul((y - np.matmul(X,theta_hat)).T,(y - np.matmul(X,theta_hat)));
    # print('var_hat',var_hat)
    temp = var_hat*np.matmul(my_xvector.T,np.matmul(np.linalg.inv(np.matmul(X.T,X)),my_xvector));
    tau = norm.ppf(alpha/2,loc = 0,scale = math.sqrt(temp.item(0)))
    # print('tau',tau)
    # print('confidence interval:',(y_pred+tau),(y_pred-tau))
    #2.7 d and e significant feature height and weight
    cov_theta = var_hat.item(0)*np.linalg.inv(np.matmul(X.T,X));
    # d = 3.841 # from chisquare tables
    cutoff = chi2.ppf(1 - alpha, df=1)

    test_stat_out = np.zeros(len(theta_hat));
    for i in range(1,len(theta_hat)):
        test_stat_out[i] = pow(theta_hat[i]/math.sqrt(cov_theta[i,i]),2)
        if test_stat_out[i] > cutoff:
            is_sig = 'Significant';
        else:
            is_sig = 'Not Significant';
        print(Col_names[i],': theta :',theta_hat[i] ,'. Significance:',test_stat_out[i],'. This
             feature is ',is_sig)
    return var_hat


#%%
def sk_linear_regression(my_xvector,df,fields):
    print('Using Scikit linearmodel regression')
    y = df[fields[0]].values
    X = df[fields[2:len(fields)]].values
    reg = linear_model.LinearRegression();
    reg.fit(X,y)
    y_pred = reg.predict(my_xvector)

    print('Coefficients: \n', reg.coef_)
    print('Intercept: \n', reg.intercept_)


    print('Prediction: \n', y_pred)
    return y_pred
```

```
#%%
def LR_CV(df,n_folds,alpha,fields):
    fold_size = math.floor(len(df)/n_folds)
    list_of_dfs = [df.loc[i:i+fold_size-1,:] for i in range(0, len(df),fold_size)]
    error_count = 0;
    for i in range(n_folds):
        test_df = list_of_dfs[i]
        train_df = df.drop(df.index[test_df.index[0]:test_df.index[-1]])
        j = 0;
        for j in range(len(test_df)):
            truth = test_df.values[j][0];
            prediction, pred_lb, pred_ub = linear_regression((test_df.values[j][2:len(fields)]),
                train_df, alpha, fields);
            if(pred_lb <= truth <= pred_ub):
                # print('Me here')
                error_count += 0;
            else:
                # print('Me here')
                error_count += 1;
    CV_out = 100*(1-(error_count/len(df)));
    print('Accuracy with cross-validation =',CV_out);
    return CV_out

#%%
def LR_CV2(df, fields, n_folds):
    y = df[fields[0]].values
    X = df[fields[2:len(fields)]].values
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)
    reg = linear_model.LinearRegression()
    reg.fit(X_train, y_train)
    scores = cross_val_score(reg, X_train, y_train, cv = n_folds)
    print("mean cross validation score: {}".format(np.mean(scores)))
    print("score without cv: {}".format(reg.score(X_train, y_train)))

    y_pred = reg.predict(X_test)
    print('R2 score:',r2_score(y_test, y_pred))
    print('regressor score:',reg.score(X_test, y_test))
    print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
    print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
    print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
#%% correlation analysis
filename = 'Reduced_data_district_2.csv'
df= read_data(filename)
df = df[df.P_Enrollment_Rate.notnull()]
df.P_Enrollment_Rate = pd.to_numeric(df.P_Enrollment_Rate, errors='coerce')
df = df.dropna()
corrMatrix = df.corr()
# sn.heatmap(corrMatrix, annot=True)
sn.heatmap(corrMatrix, vmax=0.9, square=True)
plt.show()



#%% data and histograms
filename = 'Reduced_data_district_2.csv'
```

```
fields =['P_Enrollment_Rate','Percent_schools_with_boys_toilet','Percent_schools_with_playground',\
       'Percent_schools_with_drinking_water','Percent_schools_with_electricity',\
           'Percent_schools_with_Roads','Percent_classrooms_requiring_major
               repair','Percent_single_teacher_schools','FEMALE_LIT'];
df= read_data_with_fields(filename,fields)
df = df[df.P_Enrollment_Rate.notnull()]
df.P_Enrollment_Rate = pd.to_numeric(df.P_Enrollment_Rate, errors='coerce')
df = df.dropna()
fields.insert(1,'Bias')
# df.hist(alpha=0.8)
my_xvector = np.array([65,60,96,57,87,9,7,20]).reshape(1, -1);
alpha = 0.05;
corrMatrix = df.corr()
sn.heatmap(corrMatrix, annot=True)
plt.show()




#%% Predict dropout rate using Scikits Linear regression
y_out,y_lb,y_ub = linear_regression(my_xvector,df,alpha,fields)
#%% using SKlearn
# All values in percentage/ratio
y_out = sk_linear_regression(my_xvector,df,fields)
# %% PRedict with Linear regression
# y_out = linear_regression(my_xvector,df)
#%% Significant features
var_hat = LR_significance(my_xvector,df,alpha,fields)


# %% Cross validation of linear regression
n_folds = 5;
CV_out = LR_CV(df,n_folds,alpha,fields);


# %%
LR_CV2(df, fields, n_folds)
# %%
```

Appendix C Predicting district-wise enrollment rate using Support Vector Machines and Random Forest

```
#%% ME760 Project - Predicting female literacy rate from district-wise education data
#%%
import numpy as np
import pandas as pd
import csv
import math
from sklearn.svm import SVR
from sklearn.model_selection import KFold
from sklearn.model_selection import GridSearchCV
from sklearn.metrics import r2_score
from sklearn.metrics import mean_squared_error
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
import matplotlib
from matplotlib import pyplot as plt
matplotlib.style.use('ggplot')
import scipy
from sklearn.model_selection import train_test_split
from scipy.stats import pearsonr
from sklearn.ensemble import RandomForestRegressor

def read_data_with_fields(filename,fields):
df = pd.read_csv(filename,usecols = fields)
return df

filename = 'Reduced_data_district_2.csv'
fields =['FEMALE_LIT','Percent_schools_with_boys_toilet','Percent_schools_with_playground',\
'Percent_schools_with_drinking_water','Percent_schools_with_electricity','Percent_schools_with_Roads',
    'Percent_single_teacher_schools']
df= read_data_with_fields(filename,fields)

df = df.dropna()

y = df[fields[0]].values
X = df[fields[1:7]].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20, random_state=0)

sc= MinMaxScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

kf = list(KFold(n_splits=5).split(X_train,y_train))

#SVR

C_range = [0.001, 0.01, 0.1, 1, 5, 10]
gamma_range = [0.001, 0.01, 0.1, 1, 5, 10]
kernel_range = ["rbf"]

param_grid = dict(gamma=gamma_range, C=C_range, kernel=kernel_range)
```

```
grid = GridSearchCV(SVR(), param_grid=param_grid, cv=kf, n_jobs=1, verbose=1)
grid.fit(X_train, y_train)
print("The best parameters are %s with a score of %0.2f"% (grid.best_params_, grid.best_score_))

reg_svr = SVR(C=grid.best_params_['C'], gamma=grid.best_params_['gamma'])
reg_svr.fit(X_train, y_train)
y_pred = reg_svr.predict(X_test)
print('correlation:', pearsonr(y_test, y_pred))

import matplotlib.pyplot as plt
plt.ylim(20, 110)

plt.plot(reg_svr.predict(X_test))
plt.plot(y_test)

plt.scatter(y_test,reg_svr.predict(X_test))

print('correlation:', pearsonr(y_test, y_pred))

my_xvector = np.array([65,60,96,57,87,7]).reshape(1, -1);
my_xvector = sc.fit_transform(my_xvector)
y_out = reg_svr.predict(my_xvector)
y_out

#Random Forest

param_grid = {'n_estimators': [500, 700, 1000], 'max_depth': [None, 1, 2, 3], 'min_samples_split':
    [2, 3]}
grid = GridSearchCV(RandomForestRegressor(), param_grid=param_grid, cv=kf, n_jobs=1, verbose=1)
grid.fit(X_train, y_train)
print("The best parameters are %s with a score of %0.2f"% (grid.best_params_, grid.best_score_))

clf = RandomForestRegressor(n_estimators=grid.best_params_['n_estimators'],
    max_depth=grid.best_params_['max_depth'],
    min_samples_split=grid.best_params_['min_samples_split'])
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print('correlation:', pearsonr(y_test, y_pred))

import matplotlib.pyplot as plt
plt.ylim(20, 110)
plt.plot(clf.predict(X_test))
plt.plot(y_test)
plt.ylabel('female literacy rate')
labels = ['y_pred','y_true']
plt.legend(labels)
plt.scatter(y_test,clf.predict(X_test))
plt.ylabel('y_pred')
plt.xlabel('y_true')

my_xvector = np.array([65,60,96,57,87,7]).reshape(1, -1);
my_xvector = sc.fit_transform(my_xvector)
y_out = clf.predict(my_xvector)
```

Appendix D Predicting district-wise enrollment rate using Artificial Neural Networks

```
#%% ME760 Project - Predicting female literacy rate from district-wise education data
#%%
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# import the regressor
from sklearn import metrics
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score

from keras.models import Sequential
import keras
from keras.layers import Dense
from sklearn.model_selection import KFold
from sklearn.datasets import make_regression
from sklearn.preprocessing import MinMaxScaler
from numpy import array
from sklearn.preprocessing import MinMaxScaler
import matplotlib.pyplot as plt
from sklearn.metrics import mean_squared_error
from scipy.stats import pearsonr

def read_data_with_fields(filename,fields):
df = pd.read_csv(filename,usecols = fields)
return df

# data
filename = 'Reduced_data_district_2.csv'
fields =['FEMALE_LIT','Percent_schools_with_boys_toilet','Percent_schools_with_playground',\
'Percent_schools_with_drinking_water','Percent_schools_with_electricity','Percent_schools_with_Roads',
    'Percent_single_teacher_schools']
df= read_data_with_fields(filename,fields)

df = df.dropna()
y = df[fields[0]].values
X = df[fields[1:7]].values

X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.2,random_state = 1)

sc= MinMaxScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)

model = Sequential()
model.add(Dense(10, input_dim=6, activation='relu'))
model.add(Dense(20, activation='relu'))
model.add(Dense(10, activation='relu'))
model.add(Dense(5, activation='relu'))
model.add(Dense(1, activation='linear'))

keras.optimizers.Adam(lr=0.0001, beta_1=0.9, beta_2=0.999, amsgrad=False)
model.compile(loss='mean_squared_error', optimizer='RMSprop',
    metrics=['mean_absolute_percentage_error', 'mse'])

history = model.fit(X_train, y_train, epochs=2000,
    batch_size=32,validation_split=0.15,validation_data=None,verbose=1)

plt.ylim(60, 110)
plt.plot(model.predict(X_test))
plt.plot(y_test)
plt.ylim(60, 110)
plt.plot(model.predict(X_train))
plt.plot(y_train)

y_pred = np.concatenate(model.predict(X_test)).ravel().tolist()

y_true = y_test.ravel().tolist()


print(PCC = pearsonr(y_pred, y_true))

# summarize history for loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper right')
plt.show()

my_xvector = np.array([65,60,96,57,87,7]).reshape(1, -1);
my_xvector = sc.fit_transform(my_xvector)

y_out = model.predict(my_xvector)
```

References

[1] M. Kloft, F. Stiehler, Z. Zheng, N. Pinkwart, Predicting mooc dropout over weeks using machine learning methods, in: Proceedings of the EMNLP 2014 workshop on analysis of large scale social interaction in MOOCs, 2014, pp. 60–65.

[2] S. B. Kotsiantis, C. Pierrakeas, P. E. Pintelas, Preventing student dropout in distance learning using machine learning techniques, in: International conference on knowledge-based and intelligent information and engineering systems, Springer, 2003, pp. 267–274.

[3] M. Tan, P. Shao, Prediction of student dropout in e-learning program through the use of machine learning method., International Journal of Emerging Technologies in Learning 10 (1) (2015).

[4] I. Lykourentzou, I. Giannoukos, V. Nikolopoulos, G. Mpardis, V. Loumos, Dropout prediction in e-learning courses through the combination of machine learning techniques, Computers & Education 53 (3) (2009) 950–965.

[5] M. Fetler, School dropout rates, academic performance, size, and poverty: Correlates of educational reform, Educational Evaluation and Policy Analysis 11 (2) (1989) 109–116.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.