A project report for partial fulfilment of the Degree of Bachelor of Technology
In Electronics and Communication Engineering on

"SUICIDE PREDICTION USING REGRESSION AND CLASSIFICATION MODELS"

Submitted by

Avhishek Biswas, Roll – 18700316065, Year – 2019-20.
Deep Bhattacharya, Roll – 18700316058, Year – 2019-20.
Ananya Talukdar, Roll – 18700316081, Year – 2019-20.
Arijit Chowdhury, Roll – 18700316070, Year – 2019-20.

Under the supervision of
Prof. Judhajit Sanyal
Assistant Professor, Department of Electronics and Communication Engineering

Techno International Newtown
New Town, Rajarhat, Kolkata - 700156

CERTIFICATE

This is to certify that **Avhishek Biswas, Deep Bhattacharya, Ananya Talukdar and Arijit Chowdhury** of the Department of Electronics and Communication Engineering have successfully completed a project on "**SUICIDE PREDICTION USING REGRESSION CLASSIFICATION AND NEURAL NETWORK MODELS"** during their Fourth Year in B. Tech, for the session 2016-20 (under Maulana Abul Kalam Azad University of Technology) under my supervision. The project is ready for evaluation.

Supervisor **Head of the Dept.**

_____ _____

Prof. (Dr) Manabendra Maiti

Department of Electronics and Communication Engineering

ACKNOWLEDGEMENT

We would like to thank our mentor, Prof. Judhajit Sanyal for his guidance and encouragement that helped us pursue the project. We are also obliged to Prof (Dr.) Manabendra Maiti, have spent their valuable time in discussing the required concepts. Without our teachers this attempt would never have been successfully implemented, we owe them a debt of gratitude. Our parents have always been supportive. Their blessings are the roots in our endeavors. Finally, we would like to acknowledge our departmental faculty members who have taught us for first year, as each basic concept make a complete block of a specialized subject. More than the references the fact that has been necessary is the motivation to learn more, understand and explore a topic with interest and passion, which are built and aided by our teachers.

ABSTRACT

In this project we are creating prediction and classification models using different machine learning methods.

- Linear and Spline Regression
- Neural Networks
- Naïve Bayes

The dataset that is being used is from Kaggle that has been produced by the Govt. of India.

The prediction models will be able to predict the number of deaths that can occur for each different social, professional and education categories.

We created models which were trained on data between the years 2001 -2010, and consecutively tested them on data for the year 2011 and 2012.

REQUIRED LIBRARIES

Python has gathered a lot of interest recently as a choice of language for data analysis.

Can become a common language for data science and production of web-based analytics products.

Needless to say, it still has few drawbacks too:

It is an interpreted language rather than compiled language – hence might take up more CPU time.

NumPy stands for Numerical Python. The most powerful feature of NumPy is n-dimensional array. This library also contains basic linear algebra functions, Fourier transforms, advanced random number capabilities and tools for integration with other low-level languages like Fortran, C and C++.

SciPy stands for Scientific Python. SciPy is built on NumPy. It is one of the most useful libraries for variety of high-level science and engineering modules like discrete Fourier transform, Linear Algebra, Optimization and Sparse matrices.

Matplotlib for plotting vast variety of graphs, starting from histograms to line plots to heat plot. You can use Pylab feature in ipython notebook (ipython notebook –pylab = inline) to use these plotting features inline. If you ignore the inline option, then pylab converts ipython environment to an environment, very similar to Matlab. You can also use Latex commands to add math to your plot.

Pandas for structured data operations and manipulations. It is extensively used for data munging and preparation. Pandas were added relatively recently to Python and have been instrumental in boosting Python's usage in data scientist community.

Scikit-Learn . Built on NumPy, SciPy and matplotlib, this library contains a lot of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction.

```
import pandas as pd
import matplotlib.pyplotas pltimport
numpyas np
from sklearn.linear_modelimport LinearRegression
from sklearn.model_selectionimport train_test_split
```

CLEANING AND PREPARING THE DATA

One Hot Coding of Gender

```
def replace_gender(val):
        if val== "Female": return 1
        else: return 0

new_ds['Gender'] = new_ds['Gender'].apply(replace_gender)
```

Level Coding of Social-Type

```
      new_ds['Type'].unique()
array(['Seperated', 'Widowed/Widower', 'Married', 'Divorcee',
            'Never Married'], dtype=object)
new_ds['Type'].replace({
        'Seperated' :2,
        'Widowed/Widower' :4,
        'Married' :1,
             'Divorcee' :3,
        'Never Married' :0
}, inplace= True)
```

Function to Categorize data

```
def val(x):
        employed = ['Professional Activity', 'Service (Private)','Self-employed (Business activity)',
                          'Service (Government)','Public Sector Undertaking', 'Farming/Agriculture
Activity']
        values = ['Employed','Retired','Unemployed']
        if(x in employed):
                  return values[0]
        elif(x == 'Retired Person'):
                  return values[1]
        else :return values[2]
```

Cataegorizing Education Level

```
def val(x):
        if(x == 'No Education'):
                return 0
        elif(x == 'Primary'):
                return 1
        elif(x == 'Middle'):
                return 2
        elif(x == 'Matriculate/Secondary'):
                return 3
        elif(x == 'Hr. Secondary/Intermediate/Pre-Universit'):
                return 4
        elif(x == 'Diploma'):
                return 5
        elif(x == 'Graduate'):
                return 6
        elif(x == 'Post Graduate and Above'):
                return 7
        else: return 8

         cataegory= pd.Series([])
for ind,rowin ds.iterrows():
        ds.loc[ind,"Cataegory"] = val(ds.loc[ind,"Type"])
ds = ds.astype({"Cataegory": int
```

DATASETS

YEAR	CATEGORY	TOTAL-DEATHS	PROBABILITY
2011	Never Married	91989	0.226153336
2011	Married	285045	0.70077811
2011	Separated	11112	0.027318656
2011	Divorcee	3849	0.009462699
2011	Widowed/Widower	14760	0.0362872

YEAR	CATEGORY	TOTAL-DEATHS	PROBABILITY
2011	No Education	160880	0.396039604
2011	Primary	62341	0.153465347
2011	Middle	42231	0.103960396
2011	Matriculate/Secondary	36198	0.089108911
2011	Hr. Secondary	28154	0.069306931
2011	Diploma	38209	0.094059406
2011	Graduate	22121	0.054455446
2011	Post-Grad or above	16088	0.03960396

YEAR	CATEGORY	TOTAL-DEATHS	PROBABILITY
2011	Unemployed	43568	0.323159199
2011	Farming/Agriculture	14027	0.104043199
2011	Government Service	4310	0.031968788
2011	Private Sector	11172	0.082866658
2011	Self-Employed or Other Activity	61742	0.457962157

LINEAR REGRESSION

In linear regression, the relationships are modelled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models.[3] Most commonly, the conditional mean of the response given the values of the explanatory variables (or predictors) is assumed to be an affine function of those values; less commonly, the conditional median or some other quantile is used. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of the response given the values of the predictors, rather than on the joint probability distribution of all of these variables, which is the domain of multivariate analysis.
Linear regression has many practical uses. Most applications fall into one of the following two broad categories:

If the goal is prediction, or forecasting, or error reduction, linear regression can be used to fit a predictive model to an observed data set of values of the response and explanatory variables. After developing such a model, if additional values of the explanatory variables are collected without an accompanying response value, the fitted model can be used to make a prediction of the response.

If the goal is to explain variation in the response variable that can be attributed to variation in the explanatory variables, linear regression analysis can be applied to quantify the strength of the relationship between the response and the explanatory variables, and in particular to determine whether some explanatory variables may have no linear relationship with the response at all, or to identify which subsets of explanatory variables may contain redundant information
about the response.

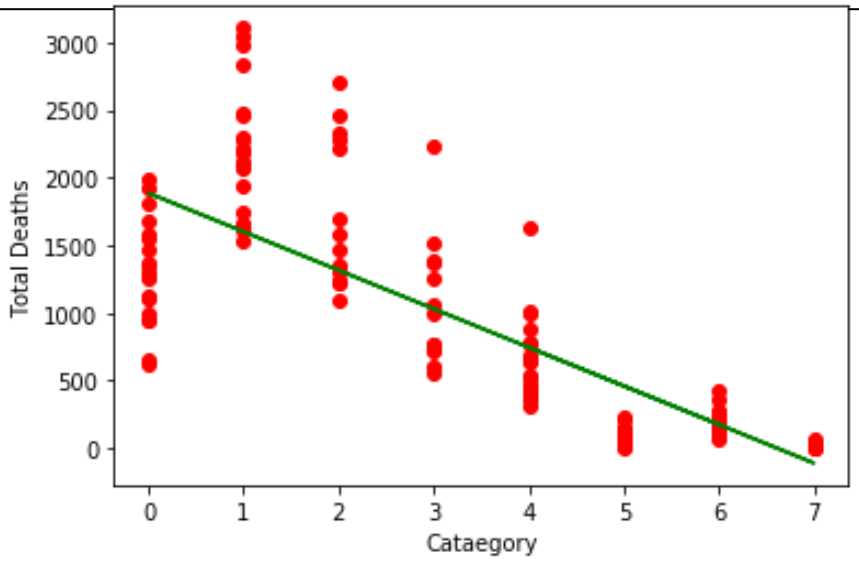
We are creating Linear Regression models on :

- Gender vs Total amount of Deaths for different Social Status
- Category for Education vs Total number of Deaths

As a result we have divided the data into training and testing .The outputs are shown in the following pages.

Creating Regression Model in Educational Status (Category vs Total)

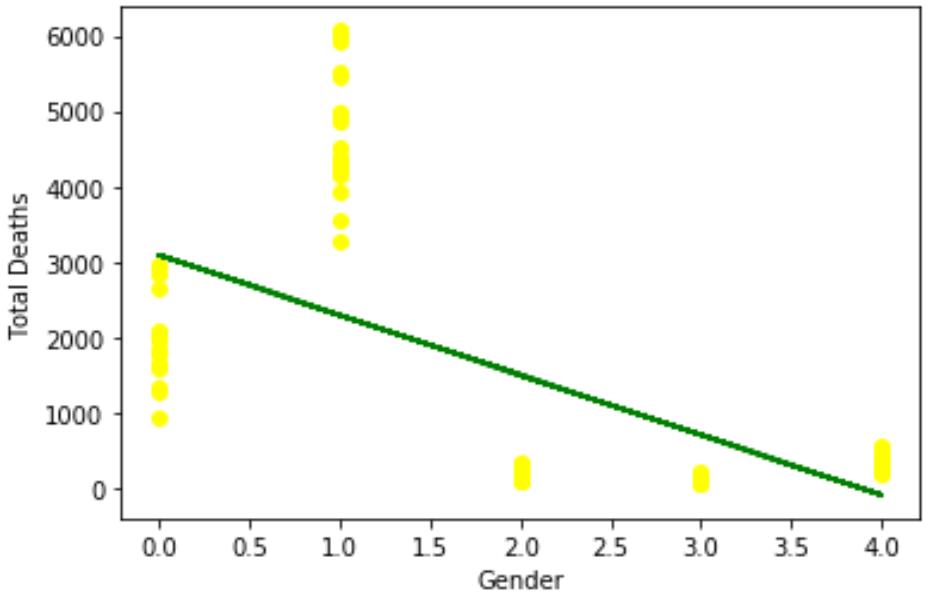
```
model = LinearRegression()
X = useful_Data.drop(['Total'],axis= 'columns') Y = useful_Data.Total
x_train, x_test, y_train, y_test= train_test_split(X,Y,test_size=1/4)
model.fit(x_train,y_train)
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normaliz e=False)
plt.scatter(x_train,y_train,color= 'red')
plt.plot(x_train,model.predict(x_train),color='green') plt.xlabel("Cataegory")
plt.ylabel("Total Deaths")
```



Linear Regression on Social Status for Gender vs Total

```
model = LinearRegression()
X = new_ds.drop(['Gender','Total'],axis = 'columns') Y = new_ds.Total

x_train, x_test, y_train, y_test= train_test_split(X,Y,test_size=1/3)
model.fit(x_train,y_train)
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normaliz e=False)
plt.scatter(x_train,y_train,color= 'yellow')
plt.plot(x_train,model.predict(x_train),color='green')
```


Total Deaths
6000
5000
4000
3000
2000
1000
0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
Gender

NAÏVE BAYES

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.
Bayes' Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes' theorem is stated
mathematically as the following equation:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

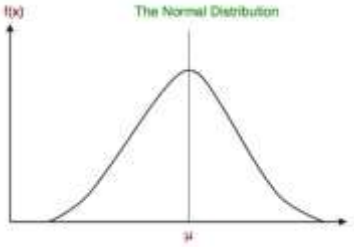
where A and B are events and P(B) > 0.

Basically, we are trying to find probability of event A, given the event B is true. Event B is also termed as evidence.
P(A) is the priori of A (the prior probability, i.e. Probability of event before evidence is seen). The evidence is an attribute value of an unknown instance(here, it is event B).
P(A|B) is a posteriori probability of B, i.e. probability of event after evidence is seen.
Now, with regards to our dataset, we can apply Bayes' theorem in following way:

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

where, y is class variable and X is a dependent feature vector (of size n)
where: X = (x_1,x_2,x_3,.....,x_n)

In Gaussian Naive Bayes, continuous values associated with each feature are assumed to be distributed according to a **Gaussian distribution**. A Gaussian distribution is also called Normal distribution. When plotted, it gives a bell shaped curve which is symmetric about the
mean of the feature values as shown below:



The likelihood of the features is assumed to be Gaussian, hence, conditional probability is given by:

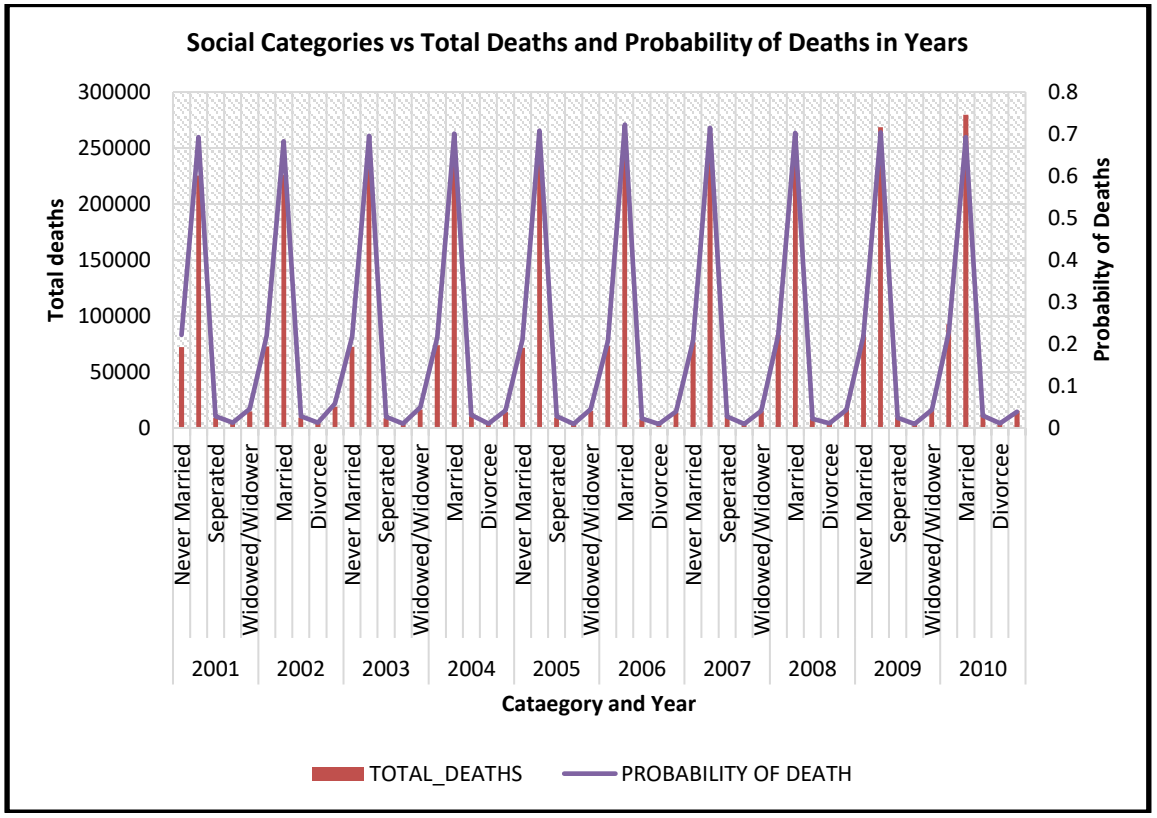
$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} exp\left(-\frac{(x_i-\mu_y)^2}{2\sigma_y^2}\right)$$

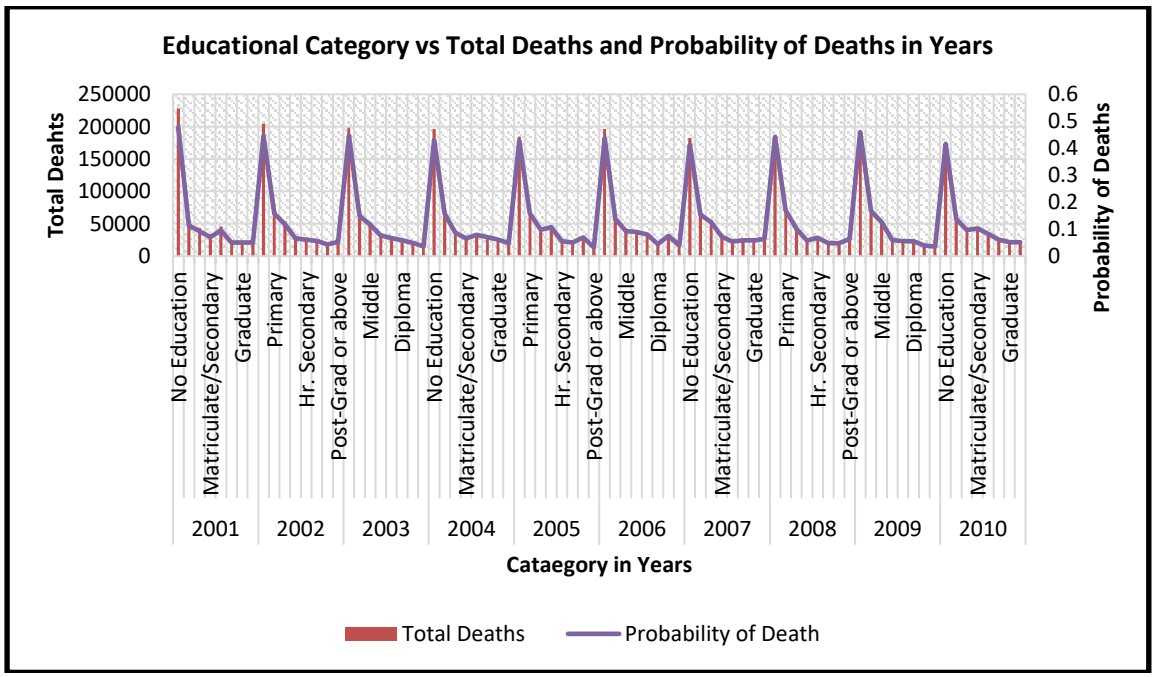
Other popular Naive Bayes classifiers are:

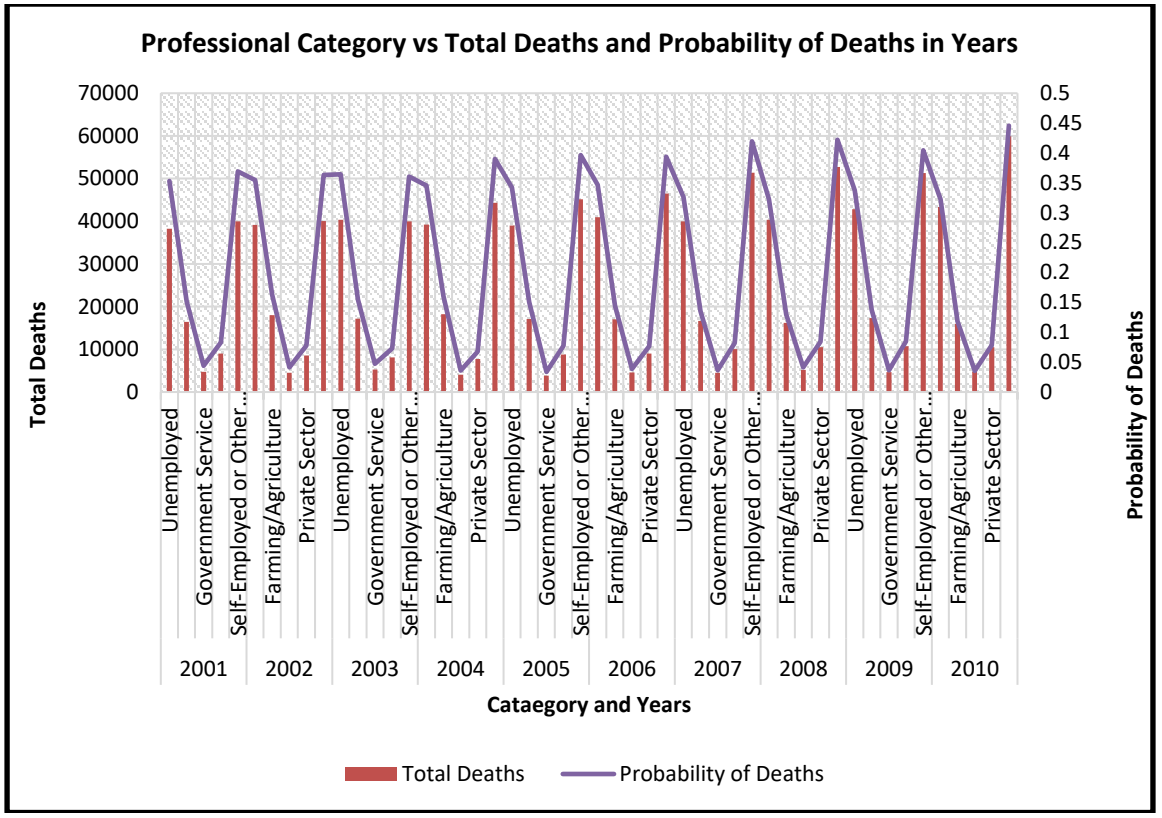
- Multinomial Naive Bayes: Feature vectors represent the frequencies with which certain events have been generated by a multinomial distribution. This is the event model typically used for document classification.

- Bernoulli Naive Bayes: In the multivariate Bernoulli event model, features are independent booleans (binary variables) describing inputs. Like the multinomial model, this model is popular for document classification tasks, where binary term occurrence(i.e. a word occurs in a document or not) features are used rather than term frequencies(i.e. frequency of a word in the document)

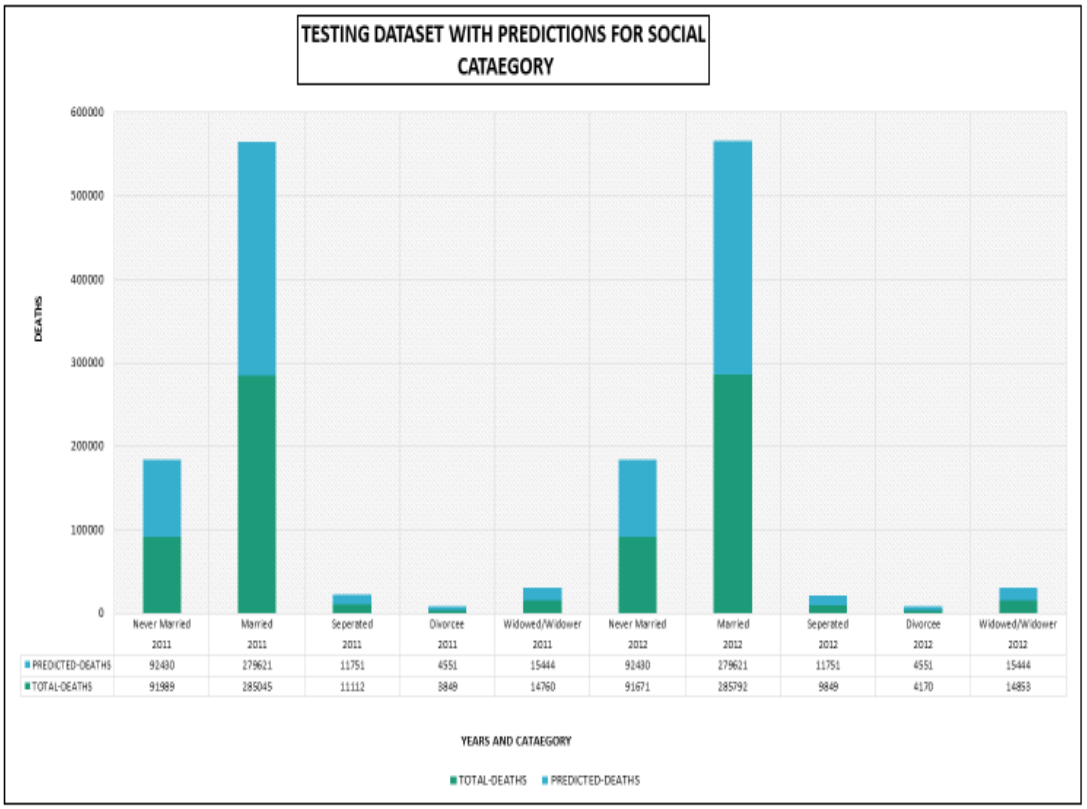
TRAINING

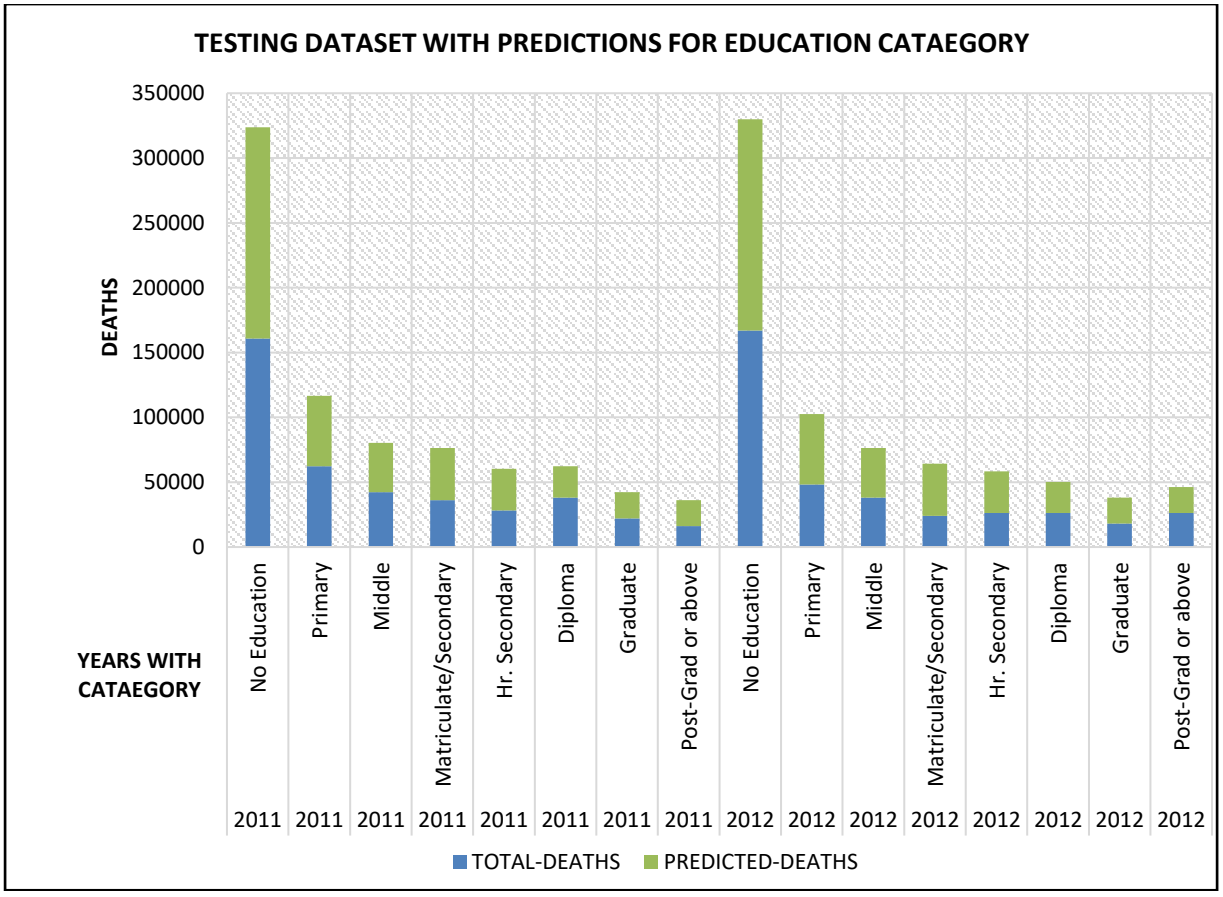

Social Categories vs Total Deaths and Probability of Deaths in Years
Total deaths
Probability of Deaths
300000
250000
200000
150000
100000
50000
0
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
Never Married
Seperated
Widowed/Widower
Married
Divorcee
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
Cataegory and Year
TOTAL_DEATHS
PROBABILITY OF DEATH

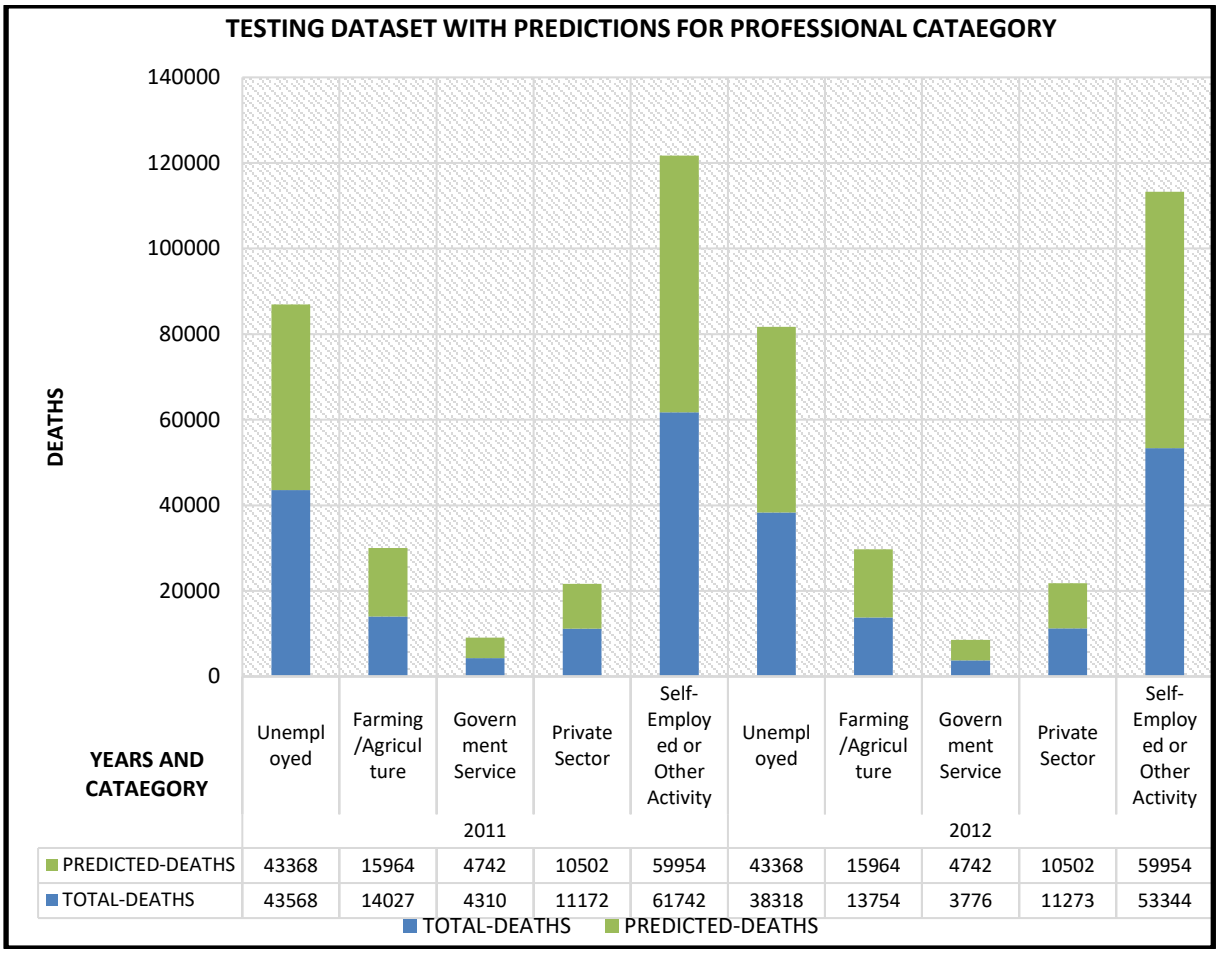

Educational Category vs Total Deaths and Probability of Deaths in Years
Total Deahts
Probability of Deaths
250000
200000
150000
100000
50000
0
0.6
0.5
0.4
0.3
0.2
0.1
0
No Education
Matriculate/Secondary
Graduate
Primary
Hr. Secondary
Post-Grad or above
Middle
Diploma
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
Cataegory in Years
Total Deaths
Probability of Death



Testing Results




TESTING DATASET WITH PREDICTIONS FOR EDUCATION CATAEGORY
DEATHS
350000
300000
250000
200000
150000
100000
50000
0
YEARS WITH CATAEGORY
No Education
Primary
Middle
Matriculate/Secondary
Hr. Secondary
Diploma
Graduate
Post-Grad or above
2011
2012
TOTAL-DEATHS
PREDICTED-DEATHS


TESTING DATASET WITH PREDICTIONS FOR PROFESSIONAL CATAEGORY
DEATHS
140000
120000
100000
80000
60000
40000
20000
0
YEARS AND CATAEGORY
Unemployed
Farming/Agriculture
Government Service
Private Sector
Self-Employed or Other Activity
2011
2012
PREDICTED-DEATHS 43368 15964 4742 10502 59954 43368 15964 4742 10502 59954
TOTAL-DEATHS 43568 14027 4310 11172 61742 38318 13754 3776 11273 53344
TOTAL-DEATHS
PREDICTED-DEATHS

CONCLUSION

The proposed method shows a high degree of accuracy in predicting the number of suicides. The method allows further refinement through cross-category joint-probabilistic suicide rate estimation.

The method allows for identification of the sub-categories of persons who are highly at risk of committing suicide. Hence, the project members hope to refine the work to provide actionable medical inputs through their endeavors.

The model can be improved in future through incorporation of classification algorithms such as SVMs and Convolutional Neural Networks.

REFERENCES

1. Berk, Richard A. (2007). "Regression Analysis: A Constructive Critique". Criminal Justice Review. 32 (3): 301–302. doi:10.1177/0734016807304871.
2. Rish, Irina. "An empirical study of the naive Bayes classifier." IJCAI 2001 workshop on empirical methods in artificial intelligence. Vol. 3. No. 22. 2001.
3. Rennie JD, Shih L, Teevan J, Karger DR. Tackling the poor assumptions of naive bayes text classifiers. InProceedings of the 20th international conference on machine learning (ICML-03) 2003 (pp. 616-623).
4. L. Jiang, H. Zhang and Z. Cai, "A Novel Bayes Model: Hidden Naive Bayes," in IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 10, pp. 1361-1371, Oct. 2009, doi: 10.1109/TKDE.2008.234.
5. Warne, Russell T. (2011). "Beyond multiple regression: Using commonality analysis to better understand R2 results". Gifted Child Quarterly. 55 (4): 313–318. doi:10.1177/0016986211422217.
6. Lange, Kenneth L.; Little, Roderick J. A.; Taylor, Jeremy M. G. (1989). "Robust Statistical Modeling Using the Distribution" (PDF). Journal of the American Statistical Association. 84 (408): 881–896. doi:10.2307/2290063. JSTOR 2290063.
7. Russell, Stuart; Norvig, Peter (2003) [1995]. Artificial Intelligence: A Modern Approach(2nd ed.). Prentice Hall. ISBN 978-0137903955.
8. Hand, D. J.; Yu, K. (2001). "Idiot's Bayes — not so stupid after all?". International Statistical Review. 69 (3): 385–399. doi:10.2307/1403452. ISSN 0306-7734. JSTOR 1403452.
9. Zhang, Harry. The Optimality of Naive Bayes (PDF). FLAIRS2004 conference.
10. Caruana, R.; Niculescu-Mizil, A. (2006). An empirical comparison of supervised learning algorithms. Proc. 23rd International Conference on Machine Learning. CiteSeerX 10.1.1.122.5901.