# VISVESVARAYA TECHNOLOGICAL UNIVERSITY

**Jnana Sangama, Belagavi - 590018**

**Mini Project Report**
**On**

# "SUICIDE RATE PREDICTION USING MACHINE LEARNING"

Submitted in partial fulfillment of the requirements for the award of the degree of

**BACHELOR OF ENGINEERING**
**In**

## ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

**by**


**SANJAY S M** **4MT20AI038**

**Under the Guidance of**
**Mrs. Vasudha G Rao**
**Assistant Professor**

**DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING**

## MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

*Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution*
(*A Unit of Rajalaxmi Education Trust®, Mangalore - 575001*)
Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi
Badaga Mijar, Moodabidri-574225, Karnataka


**2022-23**

# MANGALORE INSTITUTE OF TECHNOLOGY & ENGINEERING

*Accredited by NAAC with A+ Grade, An ISO 9001: 2015 Certified Institution*
(*A Unit of Rajalaxmi Education Trust®, Mangalore - 575001*)
Affiliated to VTU, Belagavi, Approved by AICTE, New Delhi
**Badaga Mijar, Moodabidri-574225, Karnataka**

**Department of Artificial Intelligence and Machine Learning**


## CERTIFICATE

This is to certify that the mini project entitled "**Suicide rate prediction using machine learning** " is a bonafide work carried out by **Mr. SANJAY SM (4MT20AI038)**in partial fulfillment for the requirement of 7th semester AI and ML Application Development Laboratory(18AIL76). It is certified that all the corrections/suggestions indicated for the Internal Assessment have been incorporated in the report.The mini project has been approved as it satisfies the academic requirement in respect of the 18AIL76 prescribed for the 7th Semester B.E in Artificial Intelligence And Machine Learning Program by the **Visvesvaraya Technological University,Belagavi**, for the academic year 2023 – 2024.

| | |
|---|---|
| **……** | **……** |
| Signature of the Guide | Signature of the HOD |
| **Mrs.Vasudha G Rao** | **Mr.Sunil Kumar S** |

| **Name of Examiners** | **Signature of the Examiners** |
|---|---|
| **1. …….** | **……..** |
| **2. …….** | **…….** |

# ACKNOWLDEGEMENT

The successful completion of any significant task is the outcome of invaluable aggregate combination of different people in radial direction explicitly and implicitly. We would therefore take opportunity to thank and express our gratitude to all those without whom the completion of project would not be possible

This Mini project is made under the Guidance and Project coordinator of **Mrs. Vasudha G Rao professor in department of Artificial Intelligence and Machine Learning.** We would like to express our sincere gratitude to our project guide for all the helping hand and guidance in this project

We express our sincere gratitude to **Mr. Sunil Kumar S, Senior Assistant Professor, Head of the Department of Artificial Intelligence and Machine Learning** for his support and guidance.

We would like to thank**, Dr. Prashanth C M, Principal, Mangalore Institute of Technology and Engineering, Moodabidri** for his support and encouragement

I express my sincere gratitude to our institution and management for providing us with good infrastructure, laboratory facilities, qualified and inspiring staffs, and whose guidance was of immense help in completion of this project successfully.

SANJAY SM

4MT20AI038

# ABSTRACT

This project introduces a robust suicide rate prediction system utilizing machine learning to forecast potential risks and facilitate proactive interventions in mental health. Focusing on ethical considerations, predictive analytics, and user-centric design, the system aims to address the critical societal issue of suicide prevention.

By amalgamating diverse datasets encompassing demographics, socioeconomic factors, and mental health statistics, the system harnesses the power of machine learning algorithms for predictive modeling. This enables the identification of at-risk populations or regions prone to higher suicide rates. Ethical guidelines underpin data handling, ensuring privacy and responsible usage of sensitive information.

The system's functionality includes a user-friendly interface designed for healthcare professionals and policymakers, offering intuitive access to prediction outcomes. Through interpretability features, it provides insights into the model's decision-making process, enhancing user understanding and trust.

Reliability, accuracy, and scalability constitute core pillars of system performance. It strives for real-time predictions, maintaining high accuracy levels while minimizing false positives and negatives. Scalability ensures adaptability to burgeoning data volumes and user demands without compromising performance.

Interdisciplinary collaboration with mental health experts and policymakers guides system development, ensuring relevance and effectiveness in implementing targeted interventions. Continuous maintenance, updates, and adherence to ethical standards form the bedrock of its sustainability and long-term impact.

This project represents a pivotal step toward leveraging technology ethically for societal well-being. By providing early intervention measures, the system aims to mitigate the impact of mental health challenges, fostering a proactive approach to suicide prevention and contributing to improved overall well-being in communities.

# TABLE OF CONTENTS

# LIST OF FIGURES

# Chapter 1

# INTRODUCTION

## 1.1 INTRODUCTION

Suicide is a serious public health problem. The World Health Organization (WHO) estimates that every year close to 800 000 people take their own life, which is one person every 40 seconds and there are many more people who attempt suicide. Suicide occurs throughout the lifespan and was the second leading cause of death among 15-29-yearolds globally in 2016.

Suicide does not just occur in high-income countries but is a global phenomenon in all regions of the world. In fact, over 79% of global suicides occurred in low- and middle-income countries in 2016. On average, in US there are 129 suicides per day.

The objective of this project is to predict the suicide rates using Machine Learning algorithms and to analyzing significant patterns features that result in increase of suicide rates globally.

## 1.2 DATASET DETAILS

The Dataset is borrowed from Kaggle. This is a compiled dataset pulled from four other datasets linked by time and place from year 1985 to 2016. The source of those datasets is WHO, World Bank, UNDP and a dataset published in Kaggle.

The details of the dataset are:

- **Number of Instances:** 27820
- **Number of Attributes:** 12

The below table defines attributes in the dataset:

| No. | Attribute Name | Description |
|---|---|---|
| 1 | country | Name of country |
| 2 | year | Year of the incident: 1985 to 2016 |
| 3 | sex | Gender: male or female |
| 4 | age | Range of age in years |

| | | |
|---|---|---|
| 5 | suicides_no | Number of incidents |
| 6 | population | Corresponding population of the country |
| 7 | country-year | Combination of country and year |
| 8 | HDI for year | Human development index (HDI) for year |
| 9 | gdp_for_year ($) | GDP of the country for the year |
| 10 | gdp_per_capita ($) | GDP per capita of the country for the year |
| 11 | generation | Generation of the person |
| 12 | suicides/100k pop | Number of suicides for 100k population |

## 1.3 APPLICATIONS

The application of suicide rate prediction using machine learning is versatile and can be implemented in various contexts. Here are some key applications:

- **Targeted Mental Health Services**:

Predictive models can identify specific regions or demographic groups with higher predicted suicide rates. This information can guide mental health services to allocate resources such as counselors, support groups, and mental health facilities where they're most needed.

- **Early Intervention Programs:**

Machine learning models can identify patterns indicative of potential suicidal behavior. This enables the implementation of early intervention programs, offering support and assistance to individuals at risk before a crisis occurs.

- **Policy Development:**

Predictive analytics can inform policymakers about areas or communities where suicide rates are predicted to be high. This information aids in developing targeted policies, allocating budgets, and

implementing programs focused on mental health and suicide prevention.

- **Resource Allocation:**

By accurately predicting regions or groups with a higher risk of suicide, organizations and governments can allocate financial and human resources more effectively. This ensures that preventive measures and support systems are adequately funded where they're most needed.

- **Public Awareness Campaigns:**

Understanding the factors contributing to higher predicted suicide rates can help in creating focused public awareness campaigns. These campaigns aim to reduce stigma, increase awareness about available mental health resources, and encourage help-seeking behaviors in identified high-risk areas.

- **Crisis Hotline Optimization:**

Predictive models can anticipate peak periods or regions with increased risk, allowing crisis hotlines to optimize their resources, increase staffing, and enhance support services during these critical times.

- **School Interventions**:

ML-based predictions can assist in identifying vulnerable student populations or schools at higher risk. Implementing mental health programs or interventions in these schools can provide necessary support and preventive measures to at-risk students.

- **Workplace Mental Health Initiatives:**

Predictive insights can guide companies in understanding which employee demographics or departments might be at higher risk. This information helps in developing targeted mental health programs within the workplace to support employee well-being.

- **Clinical Decision Support:**

Healthcare professionals can use predictive analytics as part of their assessments to identify individuals at risk during clinical interactions. This assists in early detection and appropriate referral to mental health services.

- **Long-term Prevention Strategies:**

By analyzing predictive patterns over time, long-term strategies can be formulated to address underlying social, economic, or mental health factors contributing to higher predicted suicide rates. This includes community-level interventions, policy changes, and ongoing support systems.

# Chapter 2

# REQUIREMENTS

## 2.1 FUNCTIONAL REQUIREMENTS

Functional requirements describe what a system or software application should do or the specific functionalities it should provide. These requirements define the actions or services that the system must perform to meet the needs of its users. They focus on the system's behaviour, features, and interactions with users or other systems. The functional requirements include:

- **Data Collection and Input**:

The system should collect data from various sources such as demographics, socioeconomic factors, mental health statistics, etc.

Accepts data in multiple formats (CSV, Excel, etc.) for input into the prediction model.

- **Data Preprocessing:**

Performs data cleaning, normalization, and feature engineering to prepare the dataset for machine learning algorithms.

Handles missing values and outliers appropriately.

- **Machine Learning Model:**

Implements a machine learning algorithm (e.g., regression, classification) capable of predicting suicide rates based on collected data.

Trains the model using historical data with relevant features.

- **Prediction Output:**

Provides a predicted suicide rate for a specified region or demographic group based on the input data.

Allows users to query and obtain predictions for specific scenarios or parameters.

- **Visualization and Reporting:**

Presents prediction results through visualizations (graphs, charts, maps) for better comprehension.

Generates reports or summaries of predicted suicide rates for different regions or timeframes.

## 2.2 NON-FUNCTIONAL REQUIREMENTS

- **Performance:**

Offers reasonably fast predictions, with an acceptable response time for queries.

Handles a significant volume of data efficiently without compromising performance.

- **Accuracy:**

Ensures the predictive model achieves a certain accuracy level based on evaluation metrics (e.g., accuracy, precision, recall, F1-score) using validation datasets.

Minimizes false positive and false negative predictions to enhance reliability.

- **Reliability:**

Maintains system availability, aiming for high uptime during operational hours.

Implements error handling to avoid system failures and ensures data consistency.

- **Usability:**

Designs an intuitive user interface allowing easy input of parameters and retrieval of predictions.

Provides clear documentation and guidance for users on how to interpret results.

- **Scalability:**

Designs the system to handle a growing dataset and user base.

Allows for scalability to accommodate increased computational demands as the dataset or model complexity grows.

- **Privacy and Ethics**:

Ensures the privacy of sensitive data used for prediction.

Adheres to ethical guidelines in handling mental health-related data and predictions.

- **Interpretability and Transparency:**

Offers explanations for predictions, making the model's decision-making process understandable to users.

Ensures transparency in the model's features and how they contribute to predictions.

- **Maintenance and Updates:**

Designs a system that allows for easy updates and incorporation of new data without significant disruptions.

Maintains documentation and support for ongoing maintenance and improvements.

## 2.3 HARDWARE REQUIREMENTS

1. Processor: AMD Ryzen 5/Intel i5 ,2GHz machine or above
2. Main memory: 8GB RAM or more
3. Hard disk drive: 500GB

## 2.4 SOFTWARE REQUIREMENTS

1. **Python:**
   - Version: 3.6 or later
2. **NumPy:**
   - Version: Latest
   - Required for numerical operations and data handling.
3. **Seaborn:**
   - Version: Latest

4. **Matplotlib:**
   - Version: Latest
   - Optional but useful for visualizing data and results.
5. **IDE (Integrated Development Environment):**
   - Examples: Visual Studio Code, PyCharm, Jupyter Notebooks
   - An environment for writing, debugging, and running code.
6. **Operating System:**
   - Windows, Linux, or macOS
   - Dependent on the platform where the application will be deployed.

## Chapter 3

# APPROACH

The following steps are implemented to build a required supervised machine learning model to predict the suicide rate of a country:

1. Data Loading from the CSV file.
2. Understanding the data.
3. Visualizing the data.
4. Preparing the data for the model.
5. Splitting the data.
6. Modeling & training.
7. Model Evaluation.

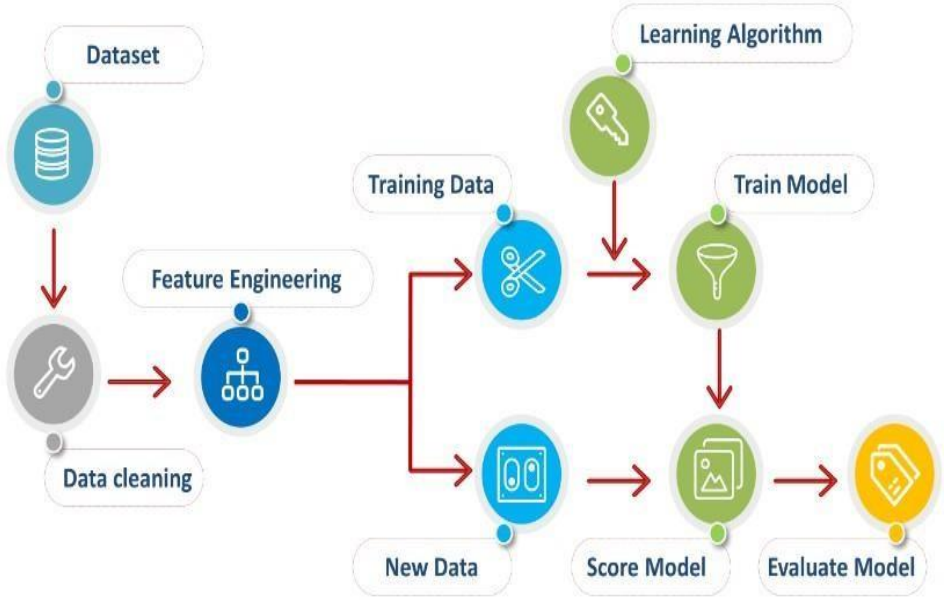


**Fig 3.1 Machine Learning Approach**

## 1. Data Loading from the CSV file

Initially, all the basic necessary libraries like Pandas, Numpy, Scikit-learn, pyplot, Seaborn etc, are imported into Jupyter Notebook. These are the main required libraries for building and training machine learning models. If any other libraries are required in the future, they can be imported accordingly.

The ~30k samples of data in the CSV file are loaded into Pandas dataframe using read_csv() function. This function returns the data in the CSV file as a two-dimensional data structure with labeled axes, called dataframe as shown below:

| | country | year | sex | age | suicides_no | population | suicides/100k pop | country-year | HDI for year | gdp_for_year ($) | gdp_per_capita ($) | generation |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | Albania | 1987 | male | 15-24 years | 21 | 312900 | 6.71 | Albania1987 | NaN | 2,156,624,900 | 796 | Generation X |
| 1 | Albania | 1987 | male | 35-54 years | 16 | 308000 | 5.19 | Albania1987 | NaN | 2,156,624,900 | 796 | Silent |
| 2 | Albania | 1987 | female | 15-24 years | 14 | 289700 | 4.83 | Albania1987 | NaN | 2,156,624,900 | 796 | Generation X |
| 3 | Albania | 1987 | male | 75+ years | 1 | 21800 | 4.59 | Albania1987 | NaN | 2,156,624,900 | 796 | G.I. Generation |
| 4 | Albania | 1987 | male | 25-34 years | 9 | 274300 | 3.28 | Albania1987 | NaN | 2,156,624,900 | 796 | Boomers |

**Fig 3.1.1: Dataframe showing few samples of data from CSV file**

## 2 Understanding the data

After successful storage of data in the dataframe, one can view the data in the tabular format and can access each part of data easily. The dataframe is of the shape *(27820, 12)*, which indicated that the number of samples and features in the dataset. Few of the give column names are renames for the convenient usage of them in this project. And the new list of the column names of the dataframe are as follows:

```
[ ]  1 #Renaming the columns names for convinience
     2 data.columns = ['country', 'year', 'gender', 'age_group', 'suicide_count',
     3                 'population', 'suicide_rate', 'country-year', 'HDI for year',
     4                 'gdp_for_year', 'gdp_per_capita', 'generation']
     5 data.columns
```

```
Index(['country', 'year', 'gender', 'age_group', 'suicide_count', 'population',
       'suicide_rate', 'country-year', 'HDI for year', 'gdp_for_year',
       'gdp_per_capita', 'generation'],
      dtype='object')
```

**Fig3.2.1: Renaming dataset columns**

**Fig3.2.2: Dataset Information & corresponding code snippet**

The following observations are made after seeing the data in the dataframe shown in Figure 3.2.1:

- Categorical features are country, year, sex, age group, country-year, generation (based on age grouping average).
- Numerical features are count of suicides, population, suicide rate, HDI for year, gdp_for_year, gdp_per_capita.
- *'HDI for year'* column has missing values. None of the other columns have any missing values. (Interpreted from Figure 3.2.1.)
- The age feature has 6 unique age group. Age is grouped into year buckets as categorical format which needs to be encoded.

```
[ ]  1 data.age_group.value_counts()
```

```
75+ years      4642
15-24 years    4642
35-54 years    4642
25-34 years    4642
55-74 years    4642
5-14 years     4610
Name: age_group, dtype: int64
```

**Fig 3.2.3: Number of Samples in each Age Group**

- Similarly, few of the categorical values needs to be encoded.
- The total number of countries in the dataset are 101 and are shown below:

```
[ ]  1 country = data.country.unique()
     2 print("Number of countries:", len(country))
     3 country
```

```
Number of countries: 101
array(['Albania', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Aruba',
       'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain',
       'Barbados', 'Belarus', 'Belgium', 'Belize',
       'Bosnia and Herzegovina', 'Brazil', 'Bulgaria', 'Cabo Verde',
       'Canada', 'Chile', 'Colombia', 'Costa Rica', 'Croatia', 'Cuba',
       'Cyprus', 'Czech Republic', 'Denmark', 'Dominica', 'Ecuador',
       'El Salvador', 'Estonia', 'Fiji', 'Finland', 'France', 'Georgia',
       'Germany', 'Greece', 'Grenada', 'Guatemala', 'Guyana', 'Hungary',
       'Iceland', 'Ireland', 'Israel', 'Italy', 'Jamaica', 'Japan',
       'Kazakhstan', 'Kiribati', 'Kuwait', 'Kyrgyzstan', 'Latvia',
       'Lithuania', 'Luxembourg', 'Macau', 'Maldives', 'Malta',
       'Mauritius', 'Mexico', 'Mongolia', 'Montenegro', 'Netherlands',
       'New Zealand', 'Nicaragua', 'Norway', 'Oman', 'Panama', 'Paraguay',
       'Philippines', 'Poland', 'Portugal', 'Puerto Rico', 'Qatar',
       'Republic of Korea', 'Romania', 'Russian Federation',
       'Saint Kitts and Nevis', 'Saint Lucia',
       'Saint Vincent and Grenadines', 'San Marino', 'Serbia',
       'Seychelles', 'Singapore', 'Slovakia', 'Slovenia', 'South Africa',
       'Spain', 'Sri Lanka', 'Suriname', 'Sweden', 'Switzerland',
       'Thailand', 'Trinidad and Tobago', 'Turkey', 'Turkmenistan',
       'Ukraine', 'United Arab Emirates', 'United Kingdom',
       'United States', 'Uruguay', 'Uzbekistan'], dtype=object)
```

**Fig3.2.4: Countries in the dataset**

# 3 Visualizing the data.

The dataset is visualized by plotting few graphs/plots using famous matplotlib and seaborn libraries. And the plots are shown below. To understand the distribution of all attributes in the given dataset, individual bar graphs are generated. The distribution graphs are shown below:

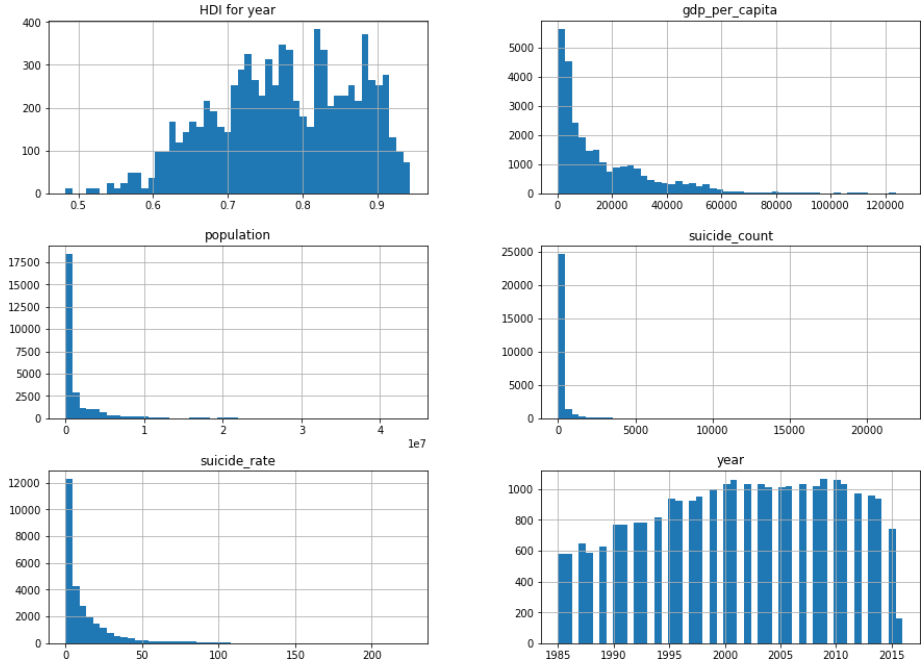


**Fig 3.3.1: Distribution graphs of features in the dataset**

To observe the relation between each attribute of the dataset, a correlation heatmap is generated and is shown below:

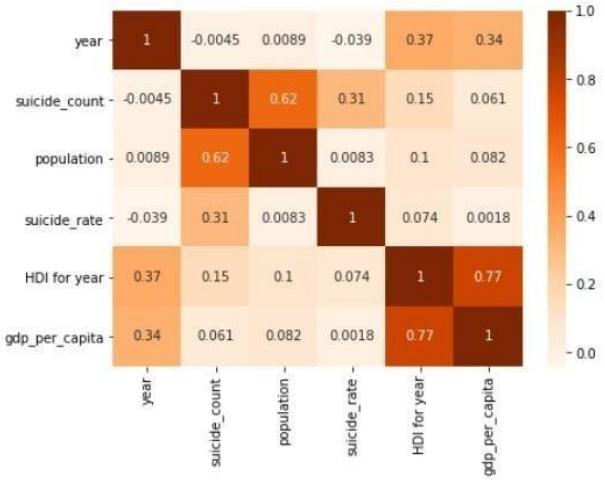

**Fig 3.3.2: Heatmap of the dataset**

The below bar plot shows the number of suicides in male and female population and we can interpret that the male population are more prone to suicide than the female.

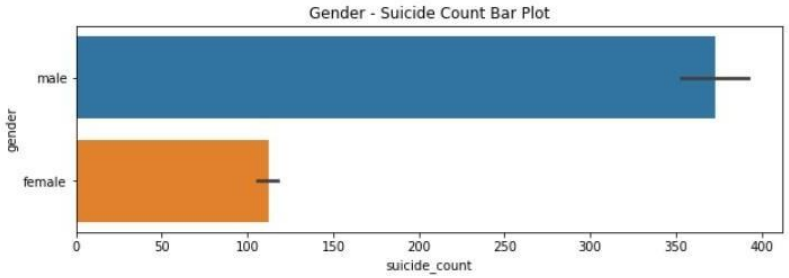


**Fig 3.3.3: Bar plot Gender – Suicide count**

Now, let's check the suicide cases based on the age group and generation separately. And the corresponding bar plots are shown below:

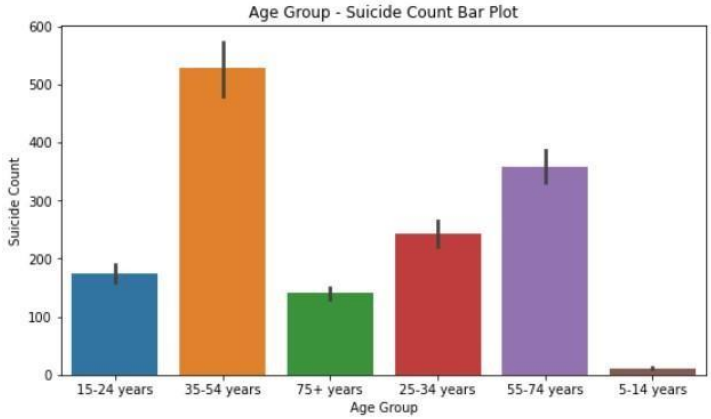


**Fig 3.3.4: Bar plot of Generation & suicide count grouped by gender**

The above boxplot shows that the suicide cases are more in the age group of 35-54 years followed by 55- 74 years. The surprising part is that the suicide cases in 5-14 year age group even though they are very less, mostly in tens. And lets see suicide count distribution in generation.

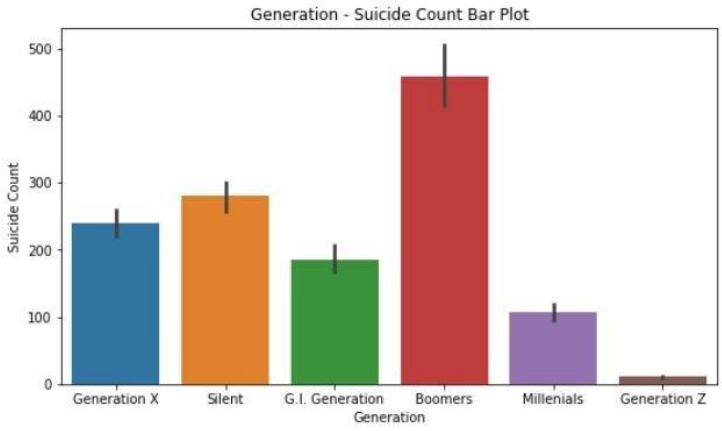


**Fig 3.3.5: Bar plot of Generation & suicide count grouped by gender**

Observation from the above plot are as below:

- The cases are more in the boomers, silent and X generations. These generations are made up of people born until 1976 based on the details provided.
- On further observation, these generations are the ones were most of them are in the age group where most suicides occur.

Now, let's see if all the above mentioned pattern exists in all the age groups, generations and also considering gender. So, the required bar plots are as show below:

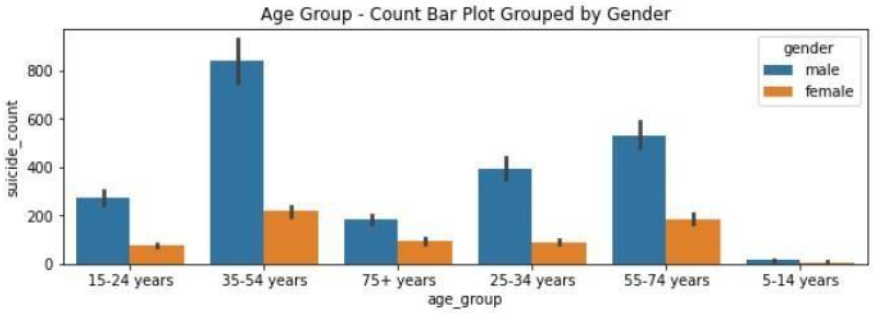


**Fig 3.3.6: Bar plot of Age group & suicide count grouped by gender**

The above bar plot stated that the 35-54 years age group is more prone to suicides irrespective of the gender followed by 55-74 years age group irrespective of the gender.

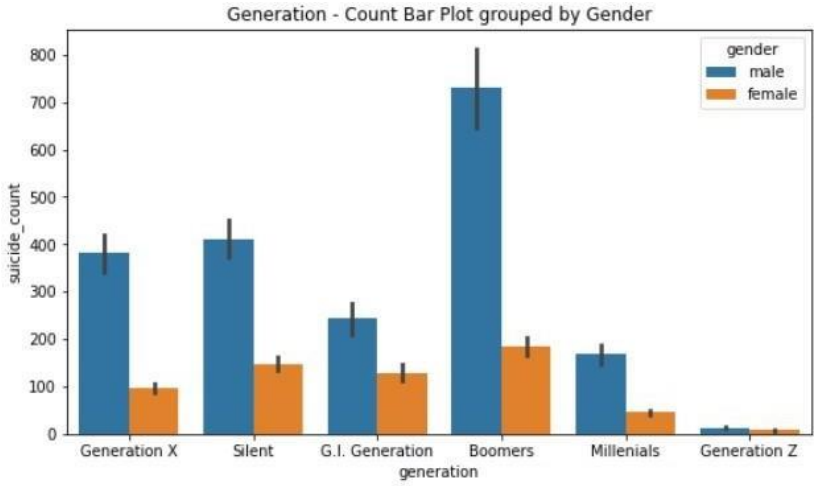


**Fig 3.3.7: Bar plot of Generation & suicide count grouped by gender**

The above bar plot stated that the Boomers generation has more suicide cases followed by Silent generation irrespective of the gender.

From the above four bar plots, it is clear that men commit suicide considerably more than women irrespective of age group and generation they belong to.

The next plot is about the countries and their suicide rate which is shown below:

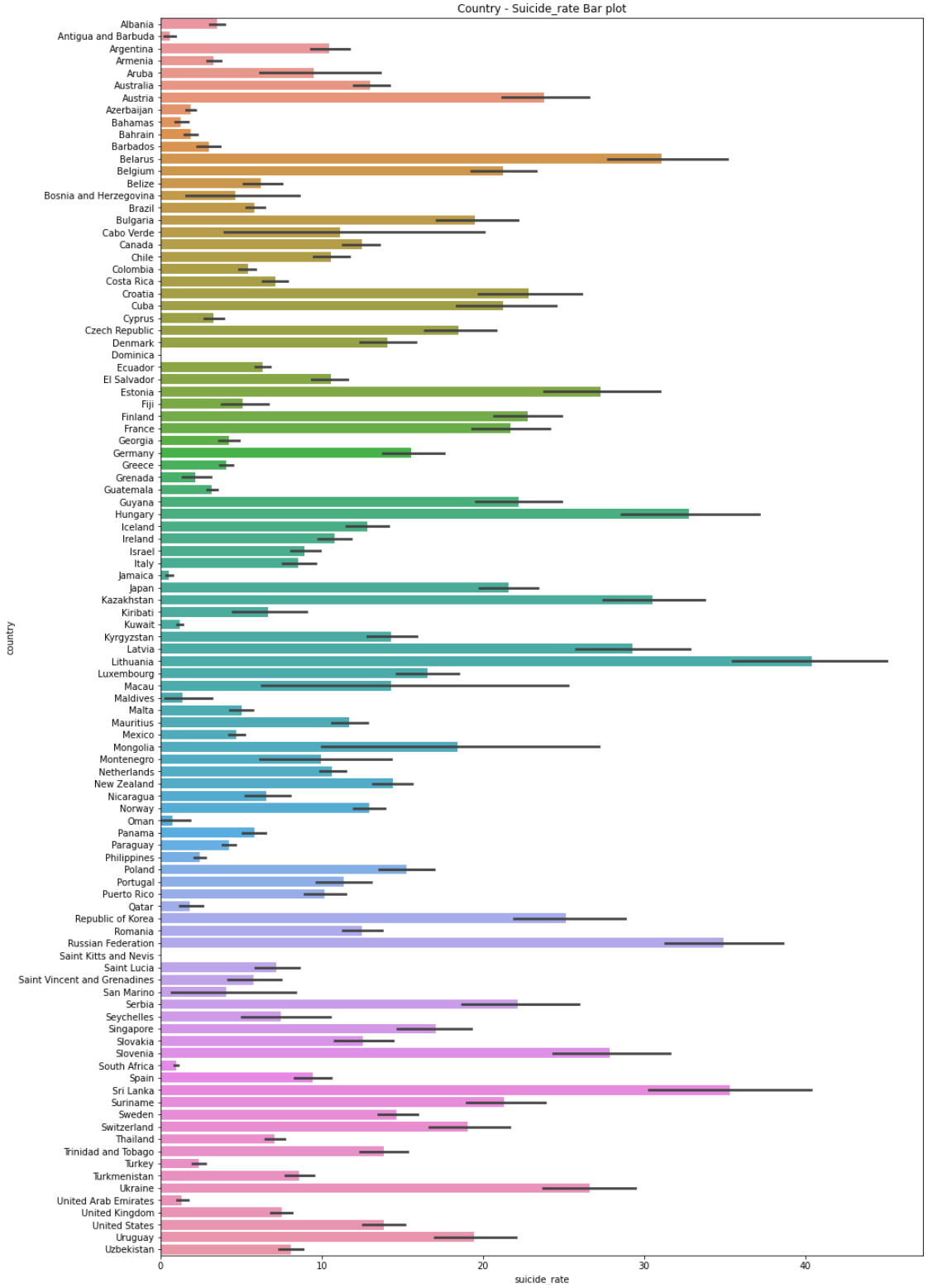


**Fig 3.3.8: Bar plot of Countries & Suicide rate**

The above bar plot shows that the highest suicide rate country is Lithuania followed by Sri Lanka.

We saw the suicide cases distribution across the countries so, lets even see it across the years. The corresponding plot is as shown below:

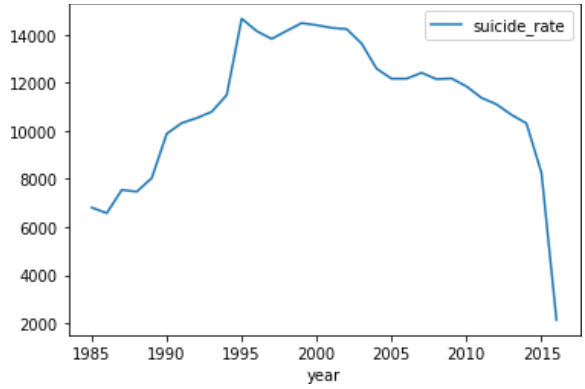


**Fig 3.3.9: Line plot of Years & Suicide rate**

The observation from the above plot are that the suicide rate had grown rapidly from year 1990 & the rate of suicide has drastically reduced in year 2016. The dataset was collected during early 2016. So, all the suicide cases of 2016 are not recorded in the dataset.

The final visualization of the dataset is its scatter matrix. This plot helps in to have a look as the outlies in the features of the data and also their distribution. The scatter matrix is shown below:

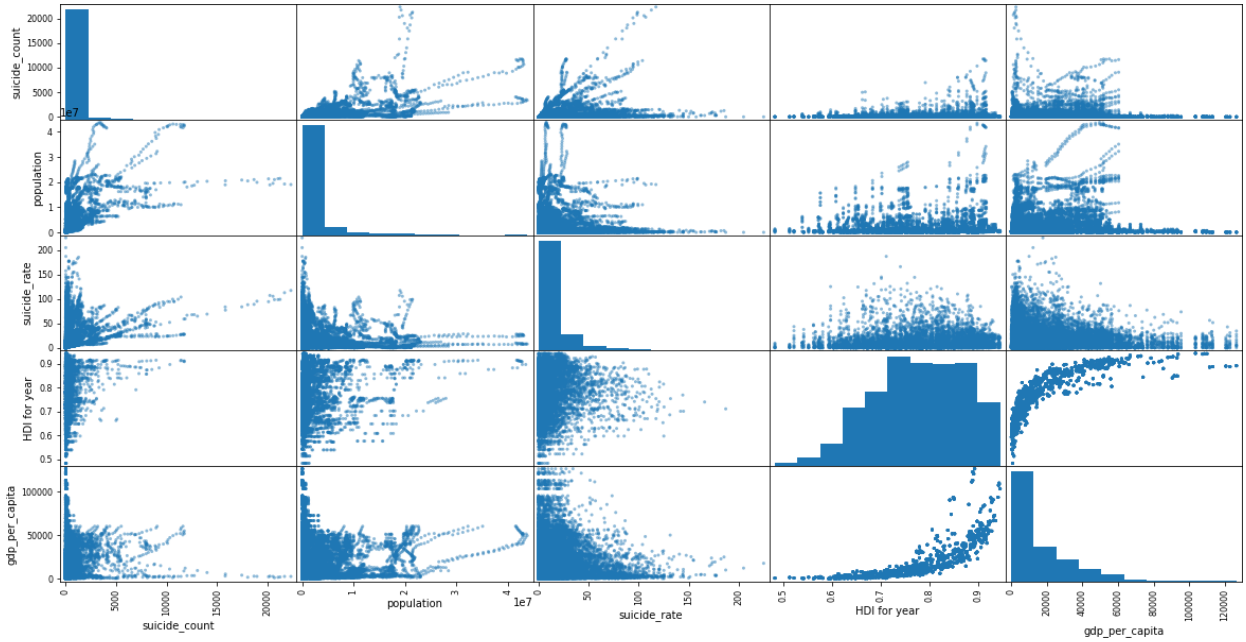


Fig 3.3.10: Scatter Matrix of Dataset

## 4 Preparing the data for the model.

In the data preprocessing step of Machine Learning, the data gets transformed, or encoded, to bring it to such a state that now the machine can easily parse it. In other words, the features of the data can now be easily interpreted by the algorithm. The steps involved in the data preprocessing are shown below:

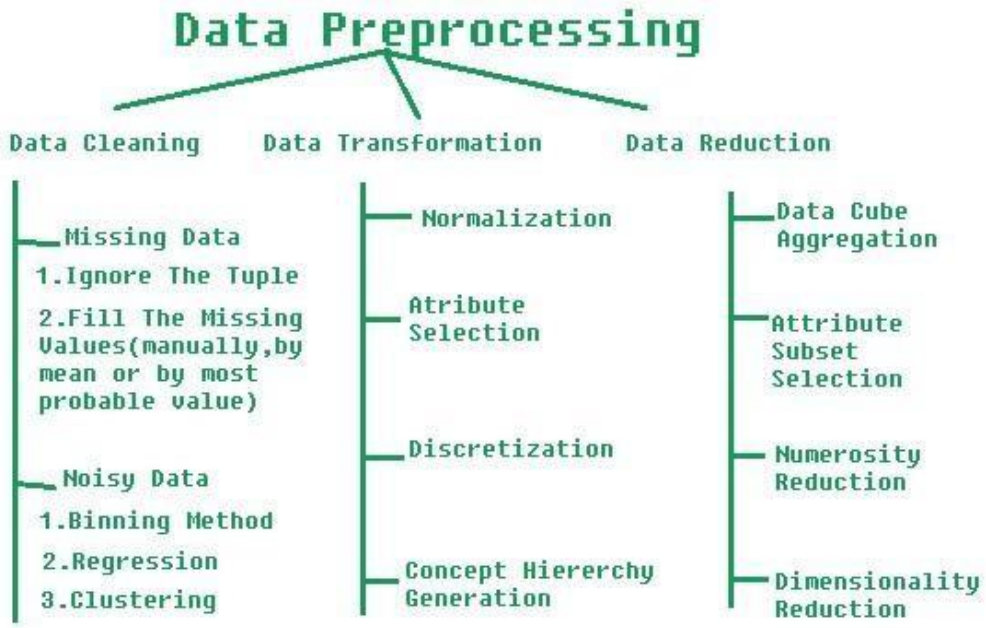


Figure 4.1: Data Preprocessing Steps

# 5 Splitting the data.

Before getting into data splitting to train and test data, we have to create feature (X) and target (y) variables from the dataset. In the suicide dataset, I considered suicide_rate of country as the target variable and the rest of them are considered as the features. The code snippet of this task is as follows:

```
[ ]  1 # Sepratating & assigning features and target columns to X & y
     2 y = data['suicide_rate']
     3 X = data.drop('suicide_rate',axis=1)
     4 X.shape, y.shape

     ((27820, 9), (27820,))
```

Figure 5.1: Creating X & y

After forming the input (X) and target (y) variables, the entire dataset needs to be split into train and test datasets. We train our model on the train data and test the accuracy of built model prediction or classification on the test data. By splitting the dataset, we are not doing any changes to the test data which gives unaltered results of our model efficiency.

The dataset splitting is done by using predefined 'train_test_split()' function from scikitlearn as shown below. And the dataset is split into 80% of train data and 20% of test data, basically an 80-20 split.

```
[ ]  1 # Splitting the dataset into train and test sets: 80-20 split
     2 from sklearn.model_selection import train_test_split
     3
     4 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 12)
     5 X_train.shape, X_test.shape

     ((22256, 9), (5564, 9))
```

Figure 5.2: Splitting the dataset

The input and target variables of train data are denoted by X_train & y_train and for test data, X_test & y_test respectively. After splitting, the training dataset has 22,256 samples and the test dataset has 5,564 samples. Now, the supervised machine learning models are trained with the training dataset in the next step.

# CHAPTER 4

## Modeling & training

Machine Learning algorithms are off two types: Supervised and Unsupervised algorithms. The type that is focused for this project is Supervised algorithms. Supervised machine learning is one of the most commonly used and successful types of machine learning. Supervised learning is used whenever we want to predict a certain outcome/label from a given set of features, and we have examples of features-label pairs.

We build a machine learning model from these features-label pairs, which comprise our training set. Our goal is to make accurate predictions for new, never-before-seen data. Further, there are two major types of supervised machine learning problems, called classification and regression. Basically, in classification, the goal is to predict a class label, which is a choice from a predefined list of possibilities.

For regression tasks, the goal is to predict a continuous number, or a floating-point number in programming terms (or real number in mathematical terms). The suicide data set comes under regression problem, as the prediction of suicide rate is a continuous number. There are wide variety of supervised machine learning algorithms or models. Among them the below mentioned models (regression) are parameter tuned, if needed and trained on the dataset.

1. k-Nearest Neighbors
2. Linear Regression
3. Decision Tree
4. Random Forest
5. Gradient Boosting
6. Multilayer Perceptrons (MLP)
7. XGBoost
8. Bagging Regressor
9. Custom Ensemble

The above mentioned parameter tune is nothing but the model hyperparameters are tuned in such a way that the model shows optimum performance on the test dataset. The beat parameters for the model can be obtained by using Scikit Learn's GridSearchCV method. This method is applied on all the algorithms.

For the evaluation of the performance of these models, the metrics considered are Accuracy & Root Mean Squared Error (RMSE). The generic process of training a model and evaluating is as follows:

- Import the model from the Scikit-Learn (if the library can be used).
- Instantiate the model and tune if the parameters if required.

- Fit the training data to the model to train it.
- The model is ready for the predictions on the test data.
- Calculate the model performance evaluation metrics.

The above mentioned five steps are applied on each model and the detailed execution is mentioned below. Before jumping into the models, I created a function to store the evaluation results of each model to a list. This step is done to make the comparison of the models easy. The code for this is as follows:

```
[ ]  1 # Creating holders to store the model performance results
     2 ML_Model = []
     3 acc_train = []
     4 acc_test = []
     5 rmse_train = []
     6 rmse_test = []
     7
     8 #function to call for storing the results
     9 def storeResults(model, a,b,c,d):
    10   ML_Model.append(model)
    11   acc_train.append(round(a, 3))
    12   acc_test.append(round(b, 3))
    13   rmse_train.append(round(c, 3))
    14   rmse_test.append(round(d, 3))
```

**Fig4.1: To Store the Model Results**

After tuning the parameters and calculating the performance of every model, the above function is called to store the training & test data accuracy & RMSE. Later this stored data is used to compare the models and determine the suitable model with set hyperparameters to the suicide dataset.

# CHAPTER 5

## Model Evaluation

The results from the models are stored in four different lists which are created before stating the model training. These lists are gathered together to form a dataframe and the code snippet is as follows:

```
[ ]  1 #creating dataframe
     2 results = pd.DataFrame({ 'ML Model': ML_Model,
     3     'Train Accuracy': acc_train,
     4     'Test Accuracy': acc_test,
     5     'Train RMSE': rmse_train,
     6     'Test RMSE': rmse_test})
```

**Fig 5.1: Custom Ensemble Model Results**

The resulting dataframe of the above execution in sorted order is as follows:

| | ML Model | Train Accuracy | Test Accuracy | Train RMSE | Test RMSE |
|---|---|---|---|---|---|
| 6 | XGBoost Regression | 0.993 | 0.988 | 0.100 | 0.134 |
| 4 | Gradient Boosted Regression | 0.988 | 0.983 | 0.130 | 0.159 |
| 7 | Bagging Regression | 0.994 | 0.982 | 0.096 | 0.166 |
| 3 | Random Forest | 0.987 | 0.980 | 0.137 | 0.176 |
| 2 | Decision Tree | 0.967 | 0.952 | 0.220 | 0.272 |
| 5 | Multilayer Perceptron Regression | 0.926 | 0.928 | 0.326 | 0.331 |
| 8 | Ensemble_SuperLearner | 0.912 | 0.910 | 0.357 | 0.371 |
| 0 | k-Nearest Neighbors Regression | 1.000 | 0.812 | 0.000 | 0.536 |
| 1 | Linear Regression | 0.288 | 0.296 | 1.013 | 1.037 |

**Fig 5.2: Models Performance Results (descending order)**

Among all the trained modesl, XGBoost performance is better. It is understandable because this model is very good in execution Speed & model performance.

## CHAPTER 6

# CONCLUSION

The suicide rate prediction system employing machine learning offers a proactive approach to mental health by forecasting potential risks. It merges data analytics with ethical considerations, aiding in early intervention and support. Through predictive models, it identifies at-risk groups or regions, enabling targeted interventions and resource allocation. Upholding ethical standards, ensuring data privacy, and collaborating with mental health experts are paramount. The user-centric design ensures an intuitive interface for healthcare professionals and policymakers. Reliability, scalability, and accuracy underpin the system's functionality, ensuring consistent performance and adaptability to increasing demands. Continuous maintenance, updates, and ethical usage of data bolster its effectiveness. Ultimately, this project represents a crucial step in leveraging technology for societal good, aiming to mitigate the impact of mental health challenges and fostering a proactive approach to suicide prevention.

# REFERENCES


1) Introduction to Machine Learning with Python by Andreas C. Müller and Sarah Guido

2) https://www.ritchieng.com/machine-learning-efficiently-search-tuning-param/

3) https://machinelearningmastery.com/xgboost-python-mini-course/

4) https://www.datavedas.com/regression-problems-in-python/

5) https://machinelearningmastery.com/super-learner-ensemble-in-python/

6) https://www.scribbr.com/statistics/statistical-tests/

7) Teachers

8) Books and internet