Final Project Report

Evaluation of Suicide Rates in the USA between 1985 to 2016

*Taslima Kotadiya*

*04/26/2020*

# Business Understanding

*As the awareness around mental health increases, we start noticing the effects and reasons of suicides. It has been previously seen that there is an expectation of lower suicide rates in developed countries due to better quality of life. Since USA is a developed country, we will use machine learning to evaluate the rate of suicides per 100,000 people as per GDP per capita over the years.*

**In this project, I will use the worldwide data for the suicides between 1986 to 2016 and evaluate it using machine learning to predict the rate of suicide for the years after 2016.**

## Business Problem To predict the suicide rate in the US after 2016 using linear regression, kNN, and Random Forest Classification method.

## Dataset

https://www.kaggle.com/russellyates88/suicide-rates-overview-1985-to-2016

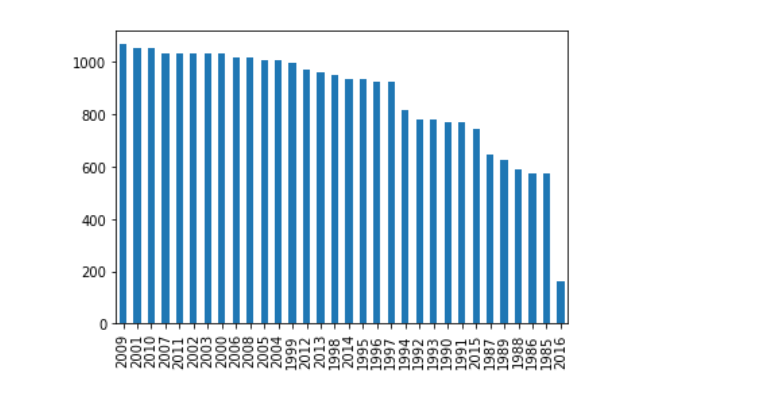
## Proposed Analytics Solution

Using linear regression, kNN, and Random Forest Classification, I will try to observe if there is a stability in the suicide rate in The USA as it is a developed country.

# Data Exploration and Preprocessing [Heading 1]

*For the chosen dataset, we have values of suicide rates from all over the world. Since we only need values for the USA, we clean the data to isolate the values and clean the data to form a new dataframe with only desired attributes.*

*While working with the initial data, we observe that the worldwide rate seems to be falling over the years since the gdp per capita rises overall.*



*Now, we run a data info check for the database and observe the headers and types of data using df.describe()*

## Data Quality Report

As we see below, we obtain the types of data

Data columns (total 11 columns):

country 27820 non-null object

year 27820 non-null int64

sex 27820 non-null object

age 27820 non-null object

suicides\_no 27820 non-null int64

population 27820 non-null int64

suicidesper100k 27820 non-null float64

HDI\_for\_year 8364 non-null float64

gdp\_for\_year 27820 non-null object

gdp\_per\_capita 27820 non-null int64

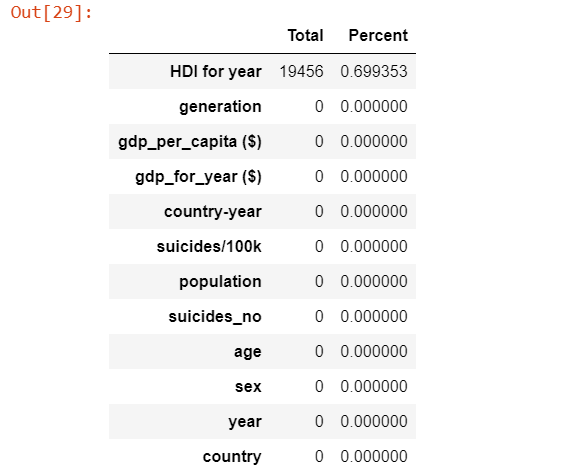
generation 27820 non-null object

dtypes: float64(2), int64(4), object(5)

Now we run a check on all the values and we see the missing values and outliers

## Missing Values and Outliers

Based on the data above, we run a check on the dataframe. We observe that there are a lot of missing values for the HDI\_for\_year value.



Therefore, we impute the values using kNN imputation with the code as below:

from sklearn.impute import KNNImputer

knn\_imputer = KNNImputer(n\_neighbors=3)

impute\_copy = df[['country', 'year', 'sex', 'age', 'suicides\_no', 'population', 'suicidesper100k', 'HDI\_for\_year','gdp\_for\_year', 'gdp\_per\_capita','generation']].copy()

imputer.fit\_transform('HDI\_for\_year')

Now, after the imputation, we start filtering the data to suit our needs.

This includes:

1. Selecting the data for USA.
2. Removing the columns that we don’t need.
3. Evaluating the statistics for the data.
4. Checking for correlations using heatmaps and scatterplots.

Details:

1. Selecting the data for USA.

To isolate the data for the USA, we use

index = df.index

index = "United States"

print(index)

df2 = df[(df['country'] == 'United States')]

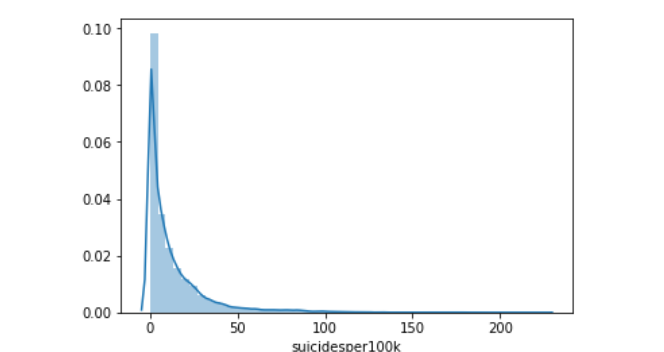
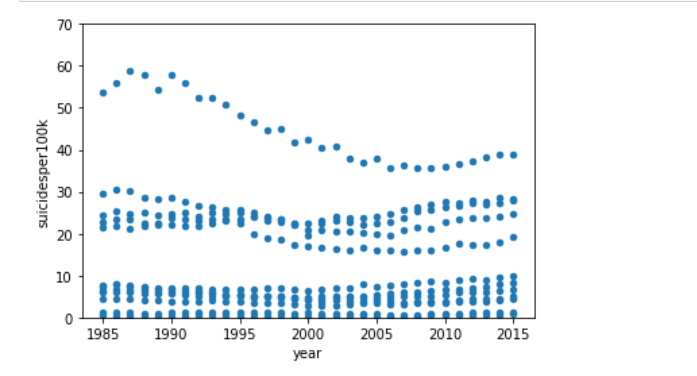
print (df2)

to transfer the data for the USA in a new temp data frame so we can do operations on it.

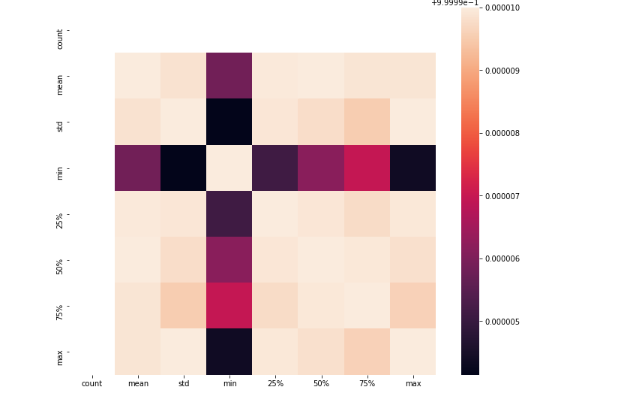
1. Removing the columns that we don’t need.

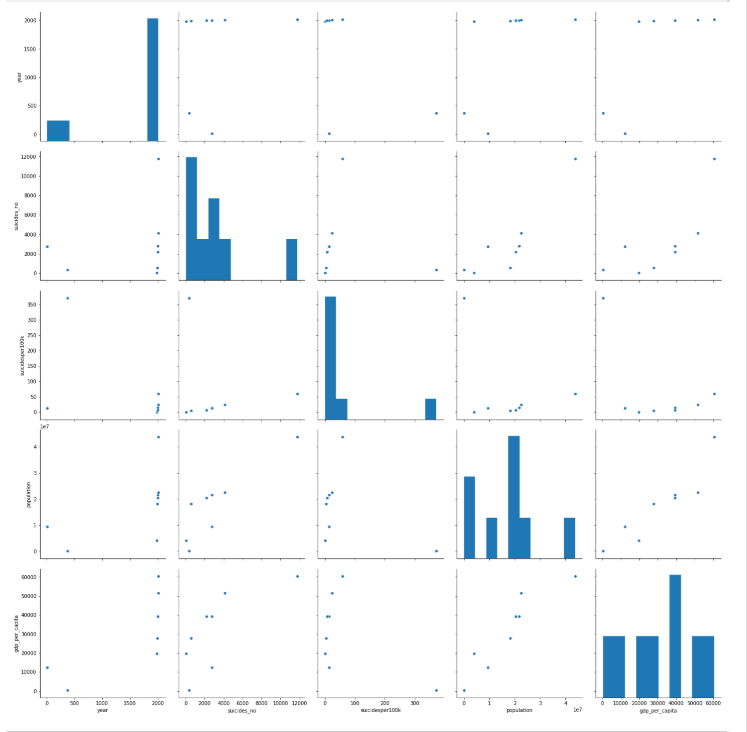
For this project, we do not need the data for HDI\_for\_year. So, we drop the column for it:

1. We run another check on the suicide rate data and observe its qualities.
2. count 27820.000000
3. mean 12.816097
4. std 18.961511
5. min 0.000000
6. 25% 0.920000
7. 50% 5.990000
8. 75% 16.620000
9. max 224.970000
10. Name: suicidesper100k, dtype: float64

1. We make scatterplots and heatmaps to see correlation in data:





## Normalization

Now we normalize data:

#normalized data

import pandas as pd

from sklearn import preprocessing

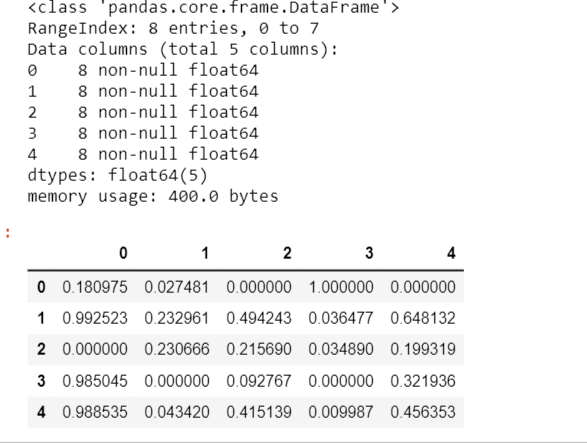
x = df4.values #returns a numpy array

min\_max\_scaler = preprocessing.MinMaxScaler()

x\_scaled = min\_max\_scaler.fit\_transform(x)

df5 = pd.DataFrame(x\_scaled)

we obtain a report of the normalized data:



## Feature Selection and Transformations [Optional]

Lorem ipsum datum ….

We run checks on the data and observe its statistical qualities such as errrors, mean, median, mode, etc.

Index: 5 entries, year to gdp\_per\_capita

Data columns (total 8 columns):

count 5 non-null float64

mean 5 non-null float64

std 5 non-null float64

min 5 non-null float64

25% 5 non-null float64

50% 5 non-null float64

75% 5 non-null float64

max 5 non-null float64

dtypes: float64(8)

Error:

Mean Absolute Error: 12.39893660360068

Mean Squared Error: 343.90644358774716

Root Mean Squared Error: 18.54471470763967

After all of the above processes, we create an Analytics Based Table for the evaluation via models and convert it to a csv file.

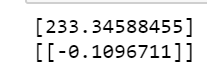
# Model Selection and Evaluation

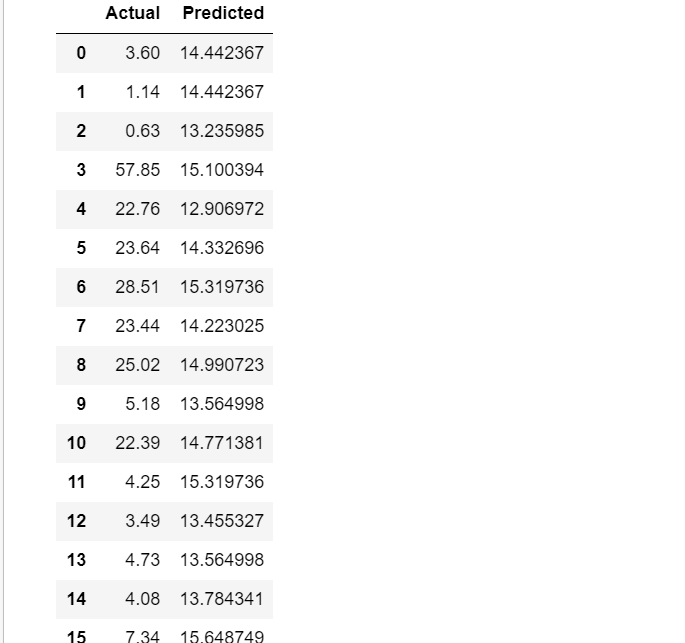
After the above process, we move on to the machine learning part of the project. In this part I’m going to use the three different types of models – linear regression, k Nearest Neighbor and Random Forest Classification to predict the future suicide rates in the USA.

The input will be the ABT and the dataset created in the above process.

## Evaluation Metrics

We calculate the evaluation metrics using the metrics from sklearn.





## Models

The models I have used are:

1. Linear regression

Using linear regression, we get a line chart that indicates a very low change in the rate of suicides. We can safely derive that the rate is constant.

1. kNN gives the ratio to be fairly similar to the other data, hence we can say the rate is again constant
2. Random forest classifier gives a decision tree that gives a predictive value as per the input data.

## Sampling and Evaluation Settings

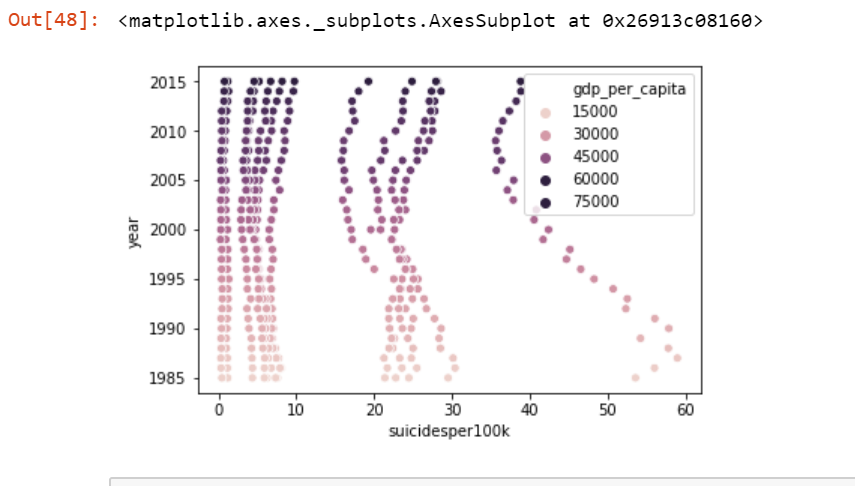
I made sure that the data was in compliance with the models. That is, I trained the model and used test data to evaluate the outcome.

## *Hyper-parameter Optimization*

Based on the obtained model, I used the value of suicides per 100k as the optimization parameter

## Evaluation

Using the above models, we see that linear regression is able to best provide the answer in this case. Since most of the values are nearly constant, it is easier to use the line chart as it shows the development in the rate which in this case is constant.



# Results and Conclusion

*From the model and the dataset, we see that linear regression has been observed to be the best model for the given data.*

*Semantically, the results show that as the GDP stays high in the USA, the rate of suicide pretty much stays constant or even reduces by a bit for that matter. Hence, we are able to say that as we had assumed, the high GDP and quality of life is a valuable factor in controlling the suicide rate.*

