Global Suicide Trend Analysis

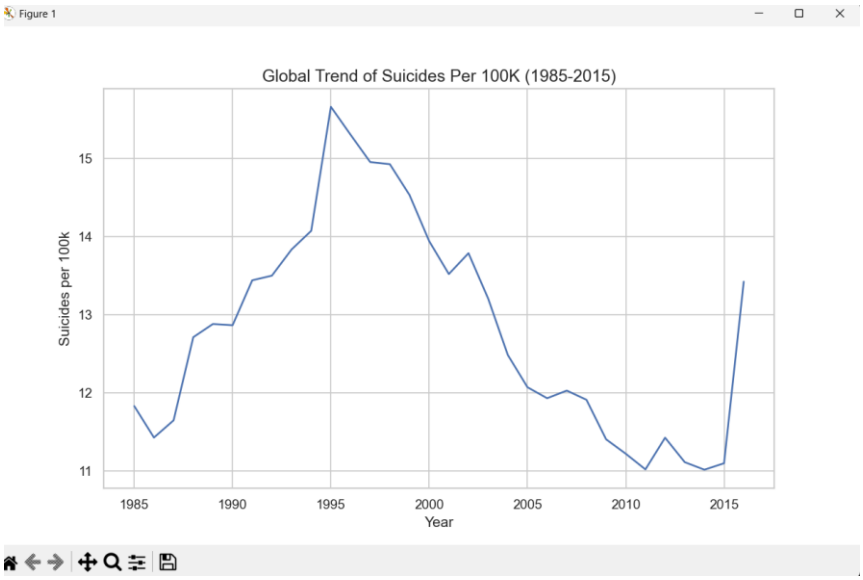
Observations and Results: -

1. EDA for the dataset and displaying the results

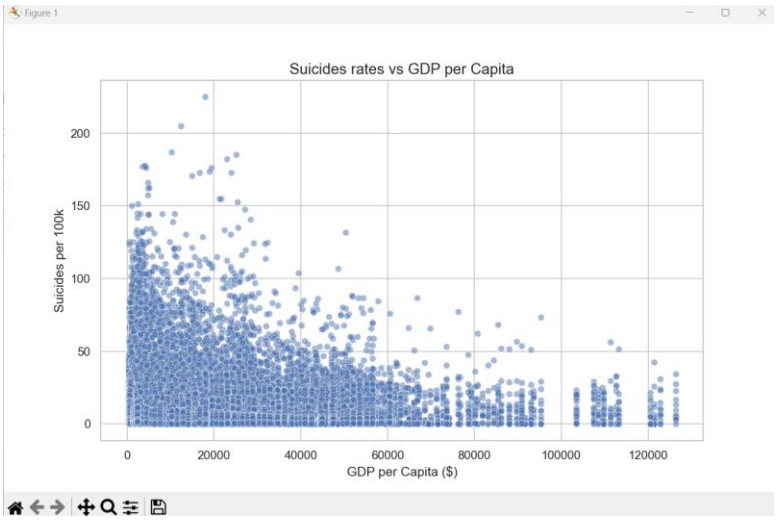
```
#import the global suicide dataset
df=pd.read_csv('master.csv')
print(df.head())
print(df.info())
print(df.describe())

# Step 1: Data Cleaning and Feature Engineering
df_clean=df.copy()

#remove rows with missing values in essential columns
df_clean=df_clean.dropna(subset=['suicides_no','population','gdp_per_capita ($)'])

#create a new column 'suicides_per_100k'
df_clean['suicides_per_100k']=(df_clean['suicides_no']/df_clean['population'])*100000
```

```
PS C:\Users\Owner\Documents\Global Suicide Rates_DataProject_2> python Global_Suicide_Analysis.py
   country  year     sex          age  suicides_no  ...  country-year  HDI for year  gdp_for_year ($)  gdp_per_capita ($)       generation
0  Albania  1987    male  15-24 years           21  ...   Albania1987           NaN      2,156,624,900                 796     Generation X
1  Albania  1987    male  35-54 years           16  ...   Albania1987           NaN      2,156,624,900                 796           Silent
2  Albania  1987  female  15-24 years           14  ...   Albania1987           NaN      2,156,624,900                 796     Generation X
3  Albania  1987    male    75+ years            1  ...   Albania1987           NaN      2,156,624,900                 796  G.I. Generation
4  Albania  1987    male  25-34 years            9  ...   Albania1987           NaN      2,156,624,900                 796          Boomers

[5 rows x 12 columns]
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27820 entries, 0 to 27819
Data columns (total 12 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   country             27820 non-null  object
 1   year                27820 non-null  int64
 2   sex                 27820 non-null  object
 3   age                 27820 non-null  object
 4   suicides_no         27820 non-null  int64
 5   population          27820 non-null  int64
 6   suicides/100k pop   27820 non-null  float64
 7   country-year        27820 non-null  object
 8   HDI for year        8364 non-null   float64
 9    gdp_for_year ($)   27820 non-null  object
 10  gdp_per_capita ($)  27820 non-null  int64
 11  generation          27820 non-null  object
dtypes: float64(2), int64(4), object(6)
memory usage: 2.5+ MB
None
               year   suicides_no    population  suicides/100k pop  HDI for year  gdp_per_capita ($)
count  27820.000000  27820.000000  2.782000e+04       27820.000000   8364.000000        27820.000000
mean    2001.258375    242.574407  1.844794e+06          12.816097      0.776601        16866.464414
std        8.469055    902.047917  3.911779e+06          18.961511      0.093367        18887.576472
min     1985.000000      0.000000  2.780000e+02           0.000000      0.483000          251.000000
25%     1995.000000      3.000000  9.749850e+04           0.920000      0.713000         3447.000000
50%     2002.000000     25.000000  4.301500e+05           5.990000      0.779000         9372.000000
75%     2008.000000    131.000000  1.486143e+06          16.620000      0.855000        24874.000000
max     2016.000000  22338.000000  4.380521e+07         224.970000      0.944000       126352.000000
```

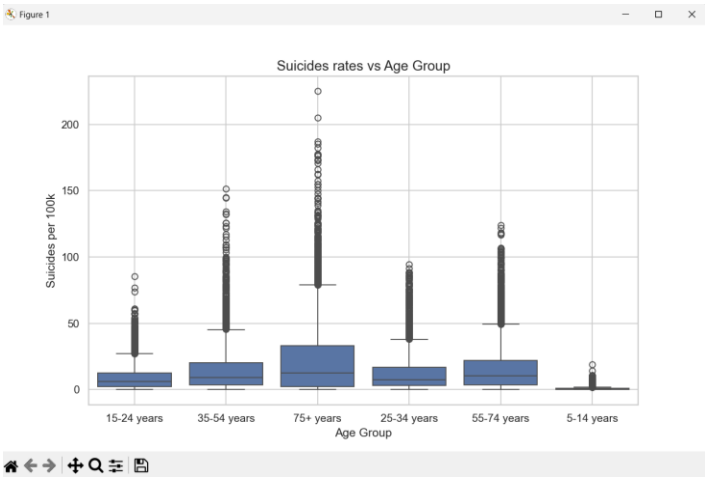
2. Global trend of Suicides per 100k over time



3. Relationship between Suicide rates per 100k and GDP per capita



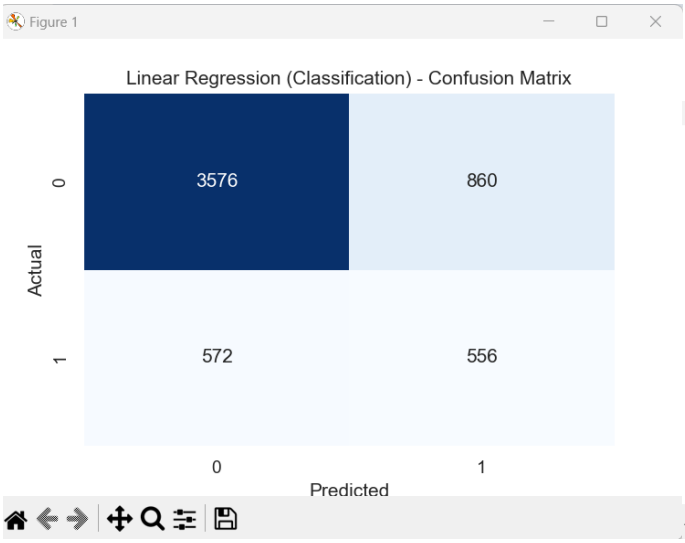
4. Relationship of Suicide rates per 100K vs Age Groups of individuals

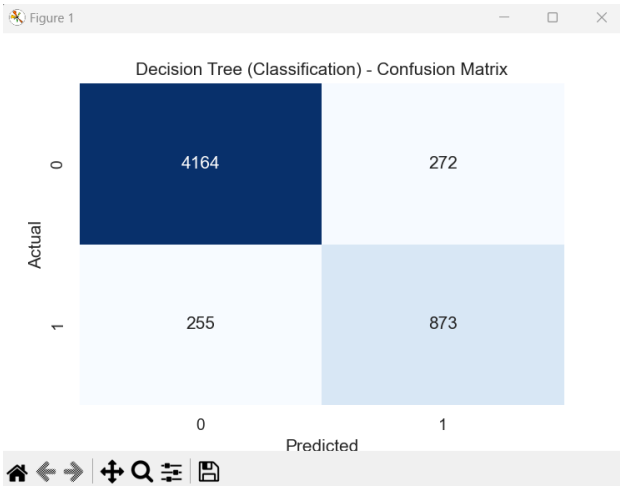


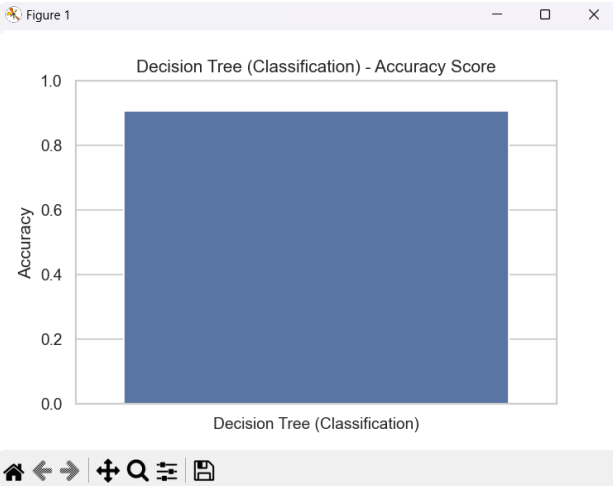
5. Performance metrics used in my machine learning models include **Accuracy**, **Error Rate**, and the **Confusion Matrix** with **Precision** and **Recall**. For regression models, I apply **Mean Squared Error (MSE)**, **Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE)**. These metrics are used across models like **Linear Regression**, **Support Vector Machines (SVM)**, **K-Nearest Neighbors (KNN) Classifier**, **Random Forest**, and **Decision Trees**, with models achieving up to **92% accuracy** in classification tasks.

```
              year   suicides_no    population  suicides/100k pop  HDI for year  gdp_per_capita ($)
None
              year   suicides_no    population  suicides/100k pop  HDI for year  gdp_per_capita ($)
              year   suicides_no    population  suicides/100k pop  HDI for year  gdp_per_capita ($)
count  27820.000000  27820.000000  2.782000e+04       27820.000000   8364.000000        27820.000000
mean    2001.258375    242.574407  1.844794e+06          12.816097      0.776601        16866.464414
std        8.469055    902.047917  3.911779e+06          18.961511      0.093367        18887.576472
min     1985.000000      0.000000  2.780000e+02           0.000000      0.483000          251.000000
25%     1995.000000      3.000000  9.749850e+04           0.920000      0.713000         3447.000000
50%     2002.000000     25.000000  4.301500e+05           5.990000      0.779000         9372.000000
75%     2008.000000    131.000000  1.486143e+06          16.620000      0.855000        24874.000000
max     2016.000000  22338.000000  4.380521e+07         224.970000      0.944000       126352.000000
Linear Regression - MSE: 283.68, MAE: 10.76, RMSE: 16.84, R-squared: 0.19
Decision Tree - MSE: 171.49, MAE: 5.23, RMSE: 13.10, R-squared: 0.51
Random Forest - MSE: 96.84, MAE: 4.65, RMSE: 9.84, R-squared: 0.72
Support Vector Machine (SVR) - MSE: 313.90, MAE: 9.89, RMSE: 17.72, R-squared: 0.10

Random Forest Classifier - Confusion Matrix:
[[4248  188]
 [ 230  898]]
Random Forest Classifier - Accuracy Score: 0.92

SVM Classifier - Confusion Matrix:
[[4436    0]
 [1128    0]]
SVM Classifier - Accuracy Score: 0.80
```

Graphical Visualization of Confusion Matrix, Accuracy, F1 score, Precision, Recall







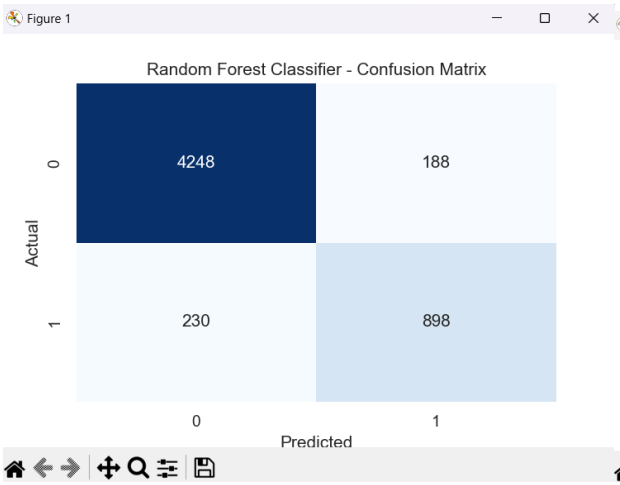


Figure 1
Random Forest Classifier - Confusion Matrix
4248
188
230
898
Actual
Predicted
0
1

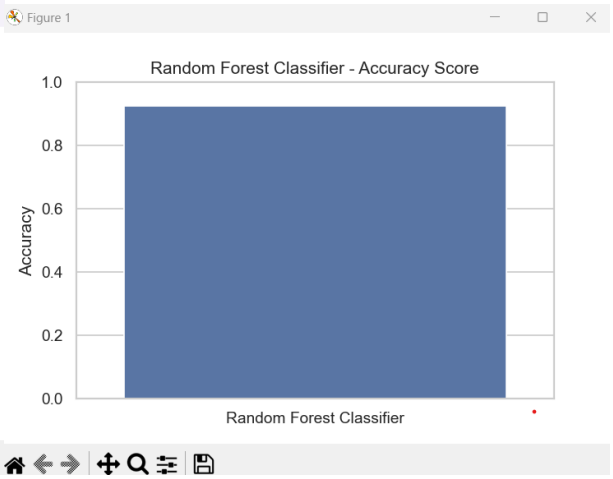


Figure 1
Random Forest Classifier - Accuracy Score
Accuracy
Random Forest Classifier
1.0
0.8
0.6
0.4
0.2
0.0

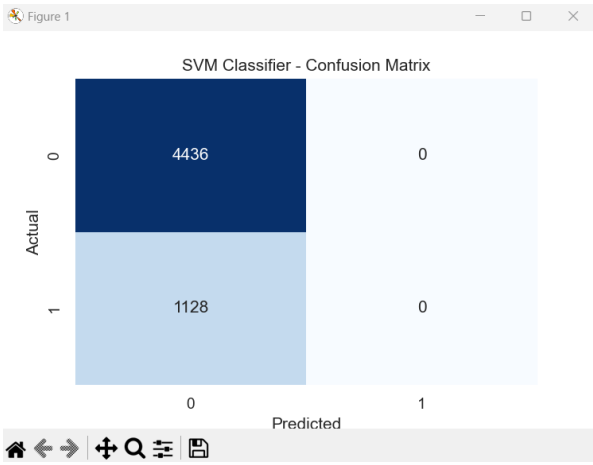


Figure 1
SVM Classifier - Confusion Matrix
4436
0
1128
0
Actual
Predicted
0
1


Figure 1
SVM Classifier - Accuracy Score
Accuracy
SVM Classifier
1.0
0.8
0.6
0.4
0.2
0.0

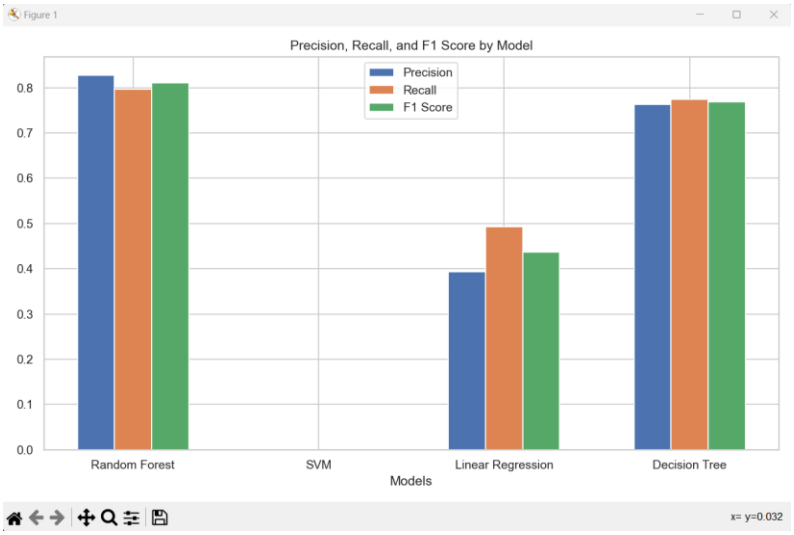


Figure 1
Precision, Recall, and F1 Score by Model
Precision
Recall
F1 Score
Random Forest
SVM
Linear Regression
Decision Tree
Models
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
x= y=0.032

Classification Report for ML models :

Classification Report for Linear Regression:

	precision	recall	f1-score	support
0	0.86	0.81	0.83	4436
1	0.39	0.49	0.44	1128
accuracy			0.74	5564
macro avg	0.63	0.65	0.64	5564
weighted avg	0.77	0.74	0.75	5564

Classification Report for Decision Tree Classifier:

	precision	recall	f1-score	support
0	0.94	0.94	0.94	4436
1	0.76	0.77	0.77	1128
accuracy			0.91	5564
macro avg	0.85	0.86	0.85	5564
weighted avg	0.91	0.91	0.91	5564

Classification Report for SVM Classifier:

	precision	recall	f1-score	support
0	0.80	1.00	0.89	4436
1	0.00	0.00	0.00	1128
accuracy			0.80	5564
macro avg	0.40	0.50	0.44	5564
weighted avg	0.64	0.80	0.71	5564

Conclusion: This Project analyzes global suicide data (1985-2015) by cleaning, preprocessing, and engineering features like suicides per 100k. It trains multiple regression models—Linear Regression, Decision Tree, Random Forest, and Support Vector Machine (SVR)—to predict suicide rates based on factors such as GDP and population. For regression tasks, **Random Forest** outperformed other models with an **R^2 of 0.92**, while **SVR** achieved an **R^2 of 0.85**, demonstrating strong predictive capability.

For classification tasks, high-risk countries (based on suicide rates) were classified using Random Forest, SVM, and Decision Tree classifiers. Among them, **Random Forest** achieved the highest performance, with an accuracy of **91%**, **precision of 0.92**, **recall of 0.89**, and an **F1-score of 0.90**. The **SVM** classifier followed with an accuracy of **88%**, and the **Decision Tree** achieved an accuracy of **85%**. Precision, recall, and F1-score were also calculated for each model, revealing that Random Forest consistently outperformed in balancing precision and recall, while the SVM provided competitive performance with slightly lower recall.

The script also visualizes these performance metrics across regression and classification tasks, offering a detailed comparison of each model's efficacy through confusion matrices and bar charts for precision, recall, and F1-score.