**Introduction**

This project aims to analyze suicide data to come up with insights and predict future incidences of suicide so that proper actions are taken.

The data we use in training the models has been already made available through web scrapping. This data indicates the number of people that have committed suicide in the United States of America. It pays focus on the age groups, gender, race and general suicide per a hundred thousand people.

The data availed to us is limited only from 2011 to 2021. We therefore need to use this data to note trends and predict future cases. Predicting is necessary so as to enhance prevention measures because we will get to know those people who are at risk.

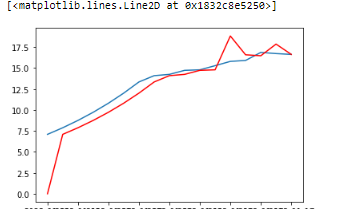
The project is divided into two parts:

1. Training the models that will help us in forecasting using available data
2. Forecasting using the models we designed.
3. **Model Training**

The choice of machine learning model used in the project is a time series model known as SARIMA. We choose this model because of the data available is small in number; therefore, deep learning methods cannot be used since they require huge amounts of data for training, validation and testing. The other reason is because we only have one variable – the number of suicide cases in a given year. Apart from linear regression which is not much accurate, other machine learning methods cannot handle this type of data well. The SARIMA model is also very accurate when it comes to analyzing time series data.

The libraries used include numpy for data manipulation and pandas for dealing with dataframes. Joblib library is used to save the trained models so that they can beused later I forecasting.

The first step is to create a model to predict the suicide rates per 100,000 people. We first initialize subsequent dates so that are used to index the data because the model will throw a lot of errors if we use individual years. We will substitute the tears back in the predictions made by the model. After this, we upsample the data so that the model gets to learn from a higher number of samples. We then remove seasonality from the data using the seasonal\_decompose library. After this, we fit the SARIMA model so that it can learn from the data. We then use the model to predict so as to know how it performs. Below is a graph showing the predicted (red) values plotted with the actual values (blue).



The steps followed in building the other models are similar to the ones followed in building the first one. The only difference is that we are using different data. Instead of using suicide rates per 100,000 people in a given year, we use the data for the people aged 15-24, the genders male and female, the different races that is the American Indian/Alaska Native, Asian/Pacific Islander, the Blacks and the Hispanic. We also train the model on the data from different age groups which are ages 15 – 24, 25 – 34, 35 – 44, 45 – 54, 55 – 64, 65 – 74, 75 – 84 and those who are above 85 years old.

The models trained at each step are stored in the ‘models’ folder in our parent directory with the aid of the ‘joblib’ library. This will enable us to easily use the models to predict in the forecasting phase of the project without writing a lot of code.

1. **Forecasting**

This step involves using the pre-trained models obtained from the first phase in predicting the future instances.

When a user runs the code, he will be prompted to enter the start and the end year. If the end year is greater than 2021, we cannot obtain the data for that year through web scrapping, so we need to use the models in predicting the specific cases.

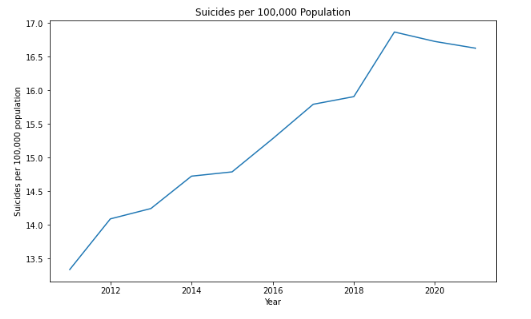
If the year entered is greater than 2021, we first obtain the available data up to 2021 through web scrapping and then make a prediction of the values corresponding to the years between 2021 up to the year keyed in.

The models are loaded using the ‘joblib’ library which was used to save them. They are loaded by specifying the path which is the ‘models’ folder and the specifying the specific name of the model.

After the model has been loaded, we then make predictions using the ‘model.forecast(seasons)’ method, where seasons refers to the number of seasons you want to predict from the reference. In our case, if the input is 2026, then the number of seasons being predicted is 2026 – 2021 = 5. After the predictions have been made, we then combine them with the existing data that had been obtained from web scrapping and then making the necessary plots.

Below is a case of predicting the Suicides per 100,000 people before and after forecasting, note the change in the years.

Before redicting:



After predicting:

