Acknowledgements

I would like to start by expressing my sincere gratitude to Dr. Abhishek Kaushik and Dr. Jack Mc Donnell, my supervisor, for their important advice, ongoing support, and patience throughout my MSc studies. My academic studies and daily life have both benefited greatly from their vast knowledge and extensive experience. Additionally, I want to thank Dr Rajesh Jaiswal and Dr Pedar Grant for their technical assistance with my work. I'd want to extend my gratitude to all the DKIT Staff members.

My time studying and living in Ireland has been excellent thanks to their generous support and assistance. Last but not least, I want to thank my brothers and sisters, and my special one. In the last several years, it wouldn't have been possible without their great support and understanding. Without their incredible support and understanding throughout the previous few years, I would not have been able to finish my research.

I am extremely thankful to all my classmates who have helped me to clear my doubts, especially Ravichandra Reddy and Satyanarayana Murthy Routhula who have been always there to spend some time outside of class to explore Ireland as well as to spend some time in the Gym helping recharge my laziness and finish writing my thesis on time.

I wish to show my appreciation to all DKIT faculties and staff for facilitating our academic year easy and efficient for us to do research and for giving us all the support and guidance.

Declaration

“I hereby declare that the work described in this project is, except where otherwise stated, entirely my own work and has not been submitted as part of any degree at this or any other Institute/University”

Signature : ___________________________

Name : ___________________________

Date : ___________________________

Dedication

“Thank God!”

To my family,

Without their love, support, and encouragement I would not be here.

To my teachers,

I am thankful to Dr Jack and Dr Abhishek from DKIT for motivating me throughout my journey of this thesis paper. Thereby, I dedicate this to them”

To my friends,

My friends around the world for all their love, support, and criticism for my personal and professional well-being.

Abstract

Objective: Analysis of the underlying reasons for suicides that occur in various nations worldwide. The main goal of this project is to create a versatile Python Dash app that can generate real-time predictions and visualizations from data provided by the backend. Make the forecasts dynamic to the data added from the backend of the admin panel. This paper explains the process of data analysis on suicide data that have occurred in various countries around the globe. Data was chosen to contain thirty-nine countries in total.

Setting: Worldwide suicide Analysis study

Subjects: Data from different countries from 1985 to 2018 including attributes like GDP Per capita, Population, Age, Gender, and Unemployment.

Main outcome measures: Models predicted the number of suicides per hundred thousand people during the following ten years. The user might select any nation from the menu to view the suicide per hundred thousand people prediction. Additionally, app users may select the number of years they wish to make a forecast for a particular period. The app itself does a model evaluation to display the amount of RMSE score that each model produces.

Results: The technique employed for this project was built on the CRISP-DM strategy. An unsupervised Machine learning Algorithm called DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is used for the POD process.

The nations with the highest rates of suicide include Russia, Ukraine, and Hungary. The age group most impacted by suicide per 100,000 people is those between the ages of 35 and 54. The study gave a bigger effect to suicide per 100,000 people than suicides total since nations with large populations may have higher suicide rates. If this is the case, the magnitude of the statistics will be inaccurate and the research's goals will not be served. As result, suicide per hundred thousand has been taken as the target variable for this research.

Dash app prediction used mainly three models. The Sarimax model showed the best results from the analysis. Followed by FB Prophet model also gave good results. Finally, SARIMAX Model seemed to show good results with less RMSE scores. My models were making live predictions each week. CRON jobs are configured on the server using the terminal as well for the model automatically every week at 10:00 AM server time. The Code used for the project is available at the following GitHub link: https://github.com/sujilkumarkm/suicide_dash_app_2022.git. The dashboard was deployed to the following 2 links: https://suicide-dash-app.herokuapp.com/ and http://204.93.172.126:8000/. Admin Panel login section is used to do CRUD(create, read, update, and delete) operations. The link for the admin panel is given here: https://www.dkit.ie.narayam.net/admin/login, here page administrators can log in with their email and password to manage the whole website which also contains a feedback section.

The link to the feedback section is https://www.dkit.ie.narayam.net/contact where page visitors can drop their feedback which will be stored in the database, and it will be available for the admins of the page to manage from the backend. This app will also send emails to user and admin separate.

Abbreviations

WHO World Health Organization

POD – Periodic Outlier Detection

ML – Machine Learning

DB – Database

SQL – Structured Query Language

VPS – Virtual Private Server

RDBMS – Relational Database Management System

DBSCAN – Density-Based Spatial Clustering of Applications with Noise

RMSE – Root Mean Squared Error

CRUD – Create, Read, Update, Delete

MVC – Model, View , Controller Paradigm

Contents

List of Tables

1 Introduction

2 Literature Review

3 Exploration of the Data

4 Design and Implementation

5 Evaluation and Testing

List of Figures

1. **Introduction**

2 **Literature Review**

3 **Exploration of the Data**

Chapter1

Introduction

This iteration is a deep dive into the suicide dataset to learn much more about the reasons for the thousands of suicides that occur each year around the world. Even though various studies on suicide have already been done previously, such as John et al. (2018), this study aimed to produce new insights that can help government bodies better grasp the problems that lie beneath them. This research could also benefit them in developing new strategies to minimize mortality rates over time. This research will look at a variety of suicide attributes and predict how many more fatalities will occur in various countries in the next years.

The goal of this research is to figure out why people commit suicide in each country. Every year, 800,000 individuals commit suicide, according to Wikipedia (2012). Suicide, for example, is becoming a more prevalent and serious problem in India, according to the World Health Organization (WHO). To address these issues, we must examine various patterns and clusters in the data and determine what circumstances cause someone to consider suicide. In addition, a web-based system will be developed that may offer dynamically illuminating visualizations of the suicide dataset, as well as opportunities for page administrators to submit new suicides to the dataset. This initiative will have a huge influence on society by allowing the government to identify and assist those who are in need, hence reducing the number of suicides each year in each nation. The government will not only save lives but also make the globe a better and safer place for people to live by implementing suitable steps based on the findings of this study.

Different social, economic, and cultural contexts exist in different countries. Russia and Ukraine, for example, are two of the most mentioned countries recently. The world is aware that the two countries are involved in a major dispute. When you see that kind of observation and data insights in Explanatory data analysis (EDA), it’s always suspicious (there is a presumption that there is a relationship (McDermott, 2016) between the conflict and suicides in two countries); the two countries, among others, have high suicide rates which need to be looked deeper to get the facts behind the figures.

Research questions	**Project Goals**
1. How suicide rates change in different countries and explain underlying issues. 2. Predict number of suicides going to happen in each continent in the next five years 3. Check for any trend in suicides in different countries over the years. 4. Check reasons why suicides in each country changed over the years? 5. Get feedbacks from page visitors and share with backend Admin panel	1. How to create a multipurpose web app for prediction. 2. How to handle mass data on the database server? 3. Check relation between GDPs Per Capita and suicide rate 4. Which country is affected by the highest number of suicides with respect to population? 5. Which age groups is more likely to suicide? 6. Check the performance of each model by looking the error and accuracy. 7. How to make Db schema? 8. Make the app work according to the update from back-end. 9. How much data is having outliers dynamically? 10. Predict top 5 countries with least number suicides in coming 5 years.

	11. How to re-run the models without affecting the speed of server or output?

Table 1.1 Research Questions and project goals

In 2011, 554 people in Ireland committed suicide, according to the CSO statistical release (2011). In terms of the country's population, this is a large figure. Each suicide will have its own set of motives, suicides like Zetzsche et al's (2007) research when they sought to figure out why people commit suicide in Western and Central Europe and came up with a few extremely interesting findings. Likewise, all the suicides that have happened in the past might be the result of a number of causes that we are not aware of. To put it differently, collecting all that information is challenging, but there are some elements that make individuals to think of committing suicide in every country. Some of these common factors are included in the suicide dataset as features, which can be used to dig deeper into the data and compare the trend in data from multiple countries.

Database design is the first step in data upload to the database. This is done using Laravel one of the most powerful and advanced PHP Framework is used. Database schema is designed using Artisan Eloquent Models. DB Table structure is defined in migration files in the Laravel ‘database’ directory.

Core technologies

• Programming Language:

– Python,

– PHP

– IDE: Visual Studio Code and Jupyter Notebook

– Libraries:

* pandas

* numpy

* requests

* sklearn
* MySQL
* dash_bootstrap_components
* dash
* matplotlib and plotly
* seaborn
* sklearn
* prophet
* statsmodels

- Frame Works
  * Bootstrap CSS,
  * PHP Laravel,
  * Python Dash
- Server
  * Heroku
  * Mochahost

Table 1.2: Core technology used in project

Python is the most used language for data analysis, according to studies from Data Camp and Bootcamp, and it is the most widely used programming language globally. Python is one of the most potent and popular programming languages in the world, so it should come as no surprise that it continues to dominate the data analysis sector. So, Python has been chosen for this study project on suicide analysis. It had taken about three months to finish this project.

• **Data Collection**: Data has been collected from the open source Kuggle Platform. Data firstly manually downloaded to the local machine then uploaded to the Jupyter notebook for further processing. A second dataset has been added to the master file which is received from a public machine learning repository. Also, to combine continents another csv was collected from internet where each country name is having continent names to classify which continent that country belongs to.

- **Data Preparation:** In this stage, Data is prepared by correcting datatypes and each row will be checked for missing or Null values and looked for outliers. These missing values and outliers are carefully treated without loss of information and keeping the accuracy and balance of the features. An output CSV is finally generated after normalizing and cleaning and that will be loaded to the MySQL database.

- **Data Exploration:** After careful preparation of data, meaningful visualisations were made using plotly, seaborn and matplotlib packages. During visualisation stage sub dataset like russia.csv was made to look deeper and understand socio-economic backgrounds and suicides relationships in few countries separately.

- **Labeling**: In order to make visualisations about the risk and non-risk groups, a new dataset with risk and non-risk columns were required. A machine learning algorithm called as DT-classifier was used to classify the suicides into risk and non-risk groups. Since final dataset contained more than hundred thousand records, doing more research about the classifier modelling was necessary to understand how this classification works.

- **Modeling**: Since the study conducted was based on sequential data, time series models were used to make predictions. SARIMA, Custom Auto Regression and FB Prophet models were created and compared before making the final app. Another machine learning algorithm was also included with modelling to make data integrity check using POD.

- **Evaluation:** Evaluation of the models is done by examining the RMSE scores. Evaluation of model is done initially on the Jupyter notebook for initial analysis as well as dynamically on the suicide dash app. These scores are used to compute goodness of the fit. It is found by taking the correlation coefficient between true values and predicted values.

- **Deployment**: The most important step in publishing a web-based application is deployment. The python dash app needs to publish on live servers like Heroku or any other popular Host service provider. So that any user could visit the app anytime. It was planned to show line plots on the number of suicides per year. MySQL database from the same service provide has been loaded and maintained with suicide dataset.

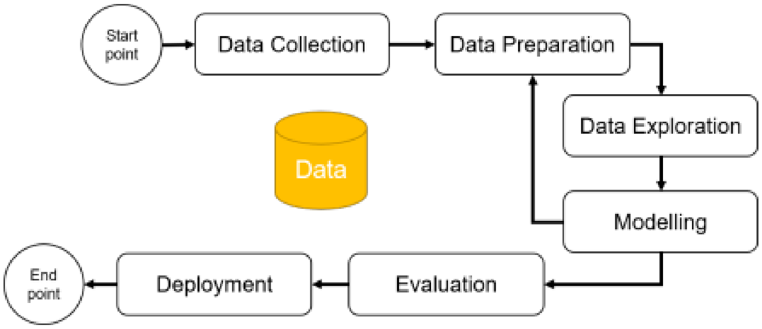


Fig 1.1 *Lifecycle of the project*

The report's framework begins with Chapter 2, which reviews the literature on the subjects most closely connected to the research questions listed in Table 1.1. Chapter 3 (Exploration of the Data) then details all procedures used to gather and clean the suicide dataset to produce the final dataset for analysis. Labeling is explained in Chapter 4 (Design and Implementation). The based model's construction and design approach for testing and training with worldwide suicides. Chapter In Chapter 5, "Parameter Evaluation, and Testing," the procedure for evaluating the based model. The number of deaths happened in Russia over time is covered in Chapter 6 (Results) in general. Chapter 7 (Conclusion) explores the question of Future work, and the research questions were addressed. Chapter 8 concludes (Deployment) details including preparation of server, connecting Git repo, migration, and DB etc.

Chapter 2

Literature Review

2.1 Related work

BI platforms like Tableau and PowerBI are excellent. It enables even non-technical managers to do their own data exploration. They are great resources for analyzing read-only datasets. The massive data science project, however, will require you to take complex and complicated operations. For example, you need to start the model retraining and activate a backend function.

Dashboards are a well-liked tool for data visualization and simple information presentation. In comparison to other forms of data visualization, dashboards have a number of benefits (Ajitesh Kumar, 2022), such as the ability to view numerous metrics at once, alter the layout to suit certain requirements, and drill down into the data for more in-depth research. One or more of the benefits of using dashboards are as follows:

I. **Aid in decision-making:** Dashboards can be used to monitor and notice patterns over time, track progress, and provide insightful data on consumer behavior and market changes.

II. **Identify business opportunities**: They offer a quick and simple approach to track progress and spot possibilities by gathering important data points and metrics.

III. **Easy to share:** Whether you embed a dashboard on a website, send information to others via email, or post it on social media, dashboards make it simple to do so. PowerPoint presentations may simply make use of dashboard screenshots.

IV. **Business and Personal use:** Dashboards can be utilized in both professional and personal settings.

Although dashboards are a popular tool for data visualization, they have several notable drawbacks (Ajitesh Kumar, 2022).

I. Dashboards can be difficult to use, especially if they attempt to include too much information. Users may struggle to identify the most crucial information and where to focus their attention.

II. Users might not be able to personalize dashboards to meet their unique demands because of how difficult it can be to do so. Due to these factors, dashboards should only be utilized when they provide the greatest means of visualizing the needed data.

III. If dashboards are not used properly, they can also be deceptive. It is simple to ignore data that contradicts an argument and cherry-pick data (data bias) that does. Confirmation bias is another name for this. Dashboards should therefore only be used sparingly and as a small component of a more comprehensive analytical strategy.

Before choosing the best framework for suicide analysis, several frameworks were considered while observing the tremendous rise of data visualization technologies. The endless and diverse advantages of various technologies were discussed in the Markus Schmitt's blog (Markus Schmitt 2022) along with comparisons of each. Streamlit and Python dash seems to have more

Several apps have been made like suicide analysis dashboard, Poverty and Equity Dashboard (Elias Dabbas, 2019) was one of the main such inspirations for creating this application. Even though there were several such works, there were no app that is dynamically updated based on the new records added from the admin panel end. Using Migration and Population Density data from World Bank. Dashboardom (Elias Dabbas, 2018) has created a dashboard. Their dashboard was designed in a simple way without having lots of CSS styles, but easily understandable for any non-technical person. They found that countries like Bahrain and Maldives have high migrant rate compared to other countries from the available world bank data. Another simple visualization project made by real python for avocados sales in US dashboard (Dylan Castillo, 2022) shows the simplest way from coding to deploying of python dash applications.

Alexander Blaufuss (Alexander Blaufuss, 2020) has created a blog on making a stock market prediction dashboard has made a simple but attractive way for the public. This work was a good example to begin with. In dashboard visualisation, 'designing' has a very important role. Making the dashboard attractive is as important as choosing right graphs for each combination of variables.

The dashboard about suicide was inspired by Johns Hopkins COVID-19 Dashboard (Johns Hopkins University, 2022). This dashboard displays real-time Covid-19 data that is regularly updated, with corresponding graphical representations created automatically. The python dashboard, which visualizes the data and insights for the public, will be the most appealing aspect of this project.

Few examples of dashboards have been discussed above; next example would be the best amongst all of them. This dashboard is from Geckoboard (Geckoboard, 2022) provides a thorough overview of corporate sales, objectives, and KPIs to sales managers. The most attractive part of dashboard is the display of most relevant information highlighted in the dashboard.

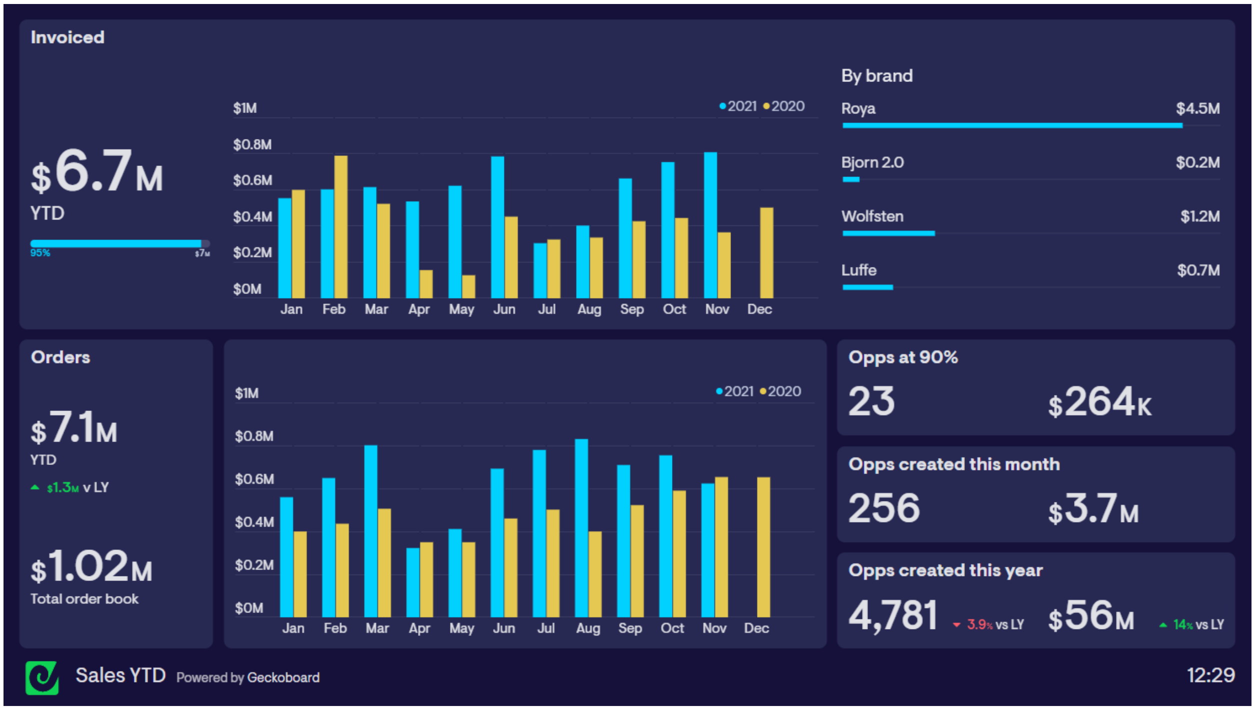


Fig 2.1 Example Dashboard from Geckoboard

Colors are carefully chosen making a common color ton for the app. Any illiterate person could easily interpret the data that has been well organized as simple graphs in this dashboard. It is simple yet powerful is the time displayed on a corner of the dash app making it easier for visitors to see changes in the trends with respect to time. Overall, the dashboard stands out with easy and meaningful insights of the data which quite relevant in dashboard making.

Geckoboard (Patowary et al., 2018) used ARIMA Model and they used dataset of natural and unnatural accidents in India between 1967 to 2015. Their study shown that ARIMA (2,2,1) model is suitable for the prediction of that dataset. In the study of Kumar Jha and Pande (Kumar Jha & Pande, 2021) found that Facebook Prophet model outperformed other models in terms of accuracy. They have used the Addictive Model and ARIMA model along with Prophet model for the forecasting. The Prophet model have shown them better fit, less error, better prediction compared to the other two. Auto Regressive AR is issued in forecast of wind speed by Huang and Chalabi (Huang & Chalabi, 1995) was a good example of how good AR Models on time series data. In this study the time varying parameters of AR model were modelled by smoothed, integrated random walk process. The whole data was utilized for the visualizations in this project. I solely utilized data from the "Russian Federation" for the time series modeling and forecasting phase. Working with time series forecasting is an important part of this dissertation. I have several different targets in this dissertation including dashboard visualization, forecast modelling, database management etc. I have been looking for ways to predict the number of suicides in upcoming years. This interest in time series and ML made me dive deep into sophisticated time series models like SARIMA and VAR to make models on the suicide data and forecast future suicides in different countries. The ARIMA model is a combination of multiple models, including the Autoregressive model, the Moving average model, and the Autoregressive Moving Average model. The form of the ARIMA model is represented by ARIMA (p, d, q), where p is the autoregressive order, d is the number of differences, and q is the moving average order.

Vector Auto Regressive Model is mostly used in finance and econometrics because they offer a framework for achieving important modelling goals, including data description, Forecasting, Structural Inference, and Policy Analysis. VAR Model is a workhouse time series multivariate model that relates current observations of a variable with past observations of itself and past observations of other variables in the system.

Thirdly, we need a database server for data to be stored on the server. I will be using PSQL or ThisSQL servers for data storage and management. I want the data in this DB to be updated from time to time and this model must be updated based on the new data injected in each time. The reason for choosing these DB’s is the flexibility of usage and its syntax matching with Structured Query Language (SQL) minute differences.

2.2 Suicide Analysis

“Data is a precious thing and will last longer than the systems themselves.”, this is a quote from Mr. Tim Berners-Lee, founder of the World Wide Web. Data is the new energy source to fuel the tech industry, despite the problems that data presents, it appears that the right use of data is bringing a lot of useful application for society. Data scientists were able to develop successful applications that could potentially aid society in understanding underlying issues that we have yet to discover with the use of effective machine learning algorithms.

What is Time Series Analysis?

A particular method of examining a set of data points gathered over a period is called as "time series analysis" (Tableau). Organizations can effectively comprehend systemic patterns across time by using time series analysis. Business users can study seasonal trends and learn more about its causes using data visualizations.

There are three types of forecasts are there(Chatfield, 2000).

(a) Judgmental forecasts: using subjective judgment, intuition, "inside" commercial knowledge, and any additional pertinent data.

(b) Univariate forecasts: which may be supplemented by a function of time like a linear trend, only consider the current and historical values of the single series being projected.

(c) Multivariate forecasts: in which predictions of a given variable are at least somewhat influenced by the values of one or more additional time series variables, sometimes known as predictor or explanatory variables. If the variables are mutually dependent, multivariate forecasts may rely on a multivariate model with several equations.

There are a wide variety of problems that time series algorithms may resolve. Companies with large sales teams employ time series forecasting to help them make better business decisions. For example, a few of these concrete examples of these potential uses are

- estimating a stock's daily closing price.
- estimating a store's daily unit sales of a product.
- predicting a state's unemployment rate each quarter.
- estimating the daily average price of gasoline.

2.4 A hope to control suicides

As suicide deaths continue to rise, it is creating serious problems for the global public healthcare system. In several nations, including Russia and Ukraine, attempts to find a solution for the suicide inclination have failed. The only way to solve this problem is by providing mental, medical, and financial support for those who in need.

(Diagnosis and Treatment 2022) : To help discover what may be triggering a person’s suicidal thoughts and to determine the appropriate treatment, also doctor may perform a medical examination, tests, and in-depth inquiries about mental and physical health of the individual.

Assessments could consist of:

Mental Health Conditions: Suicidal thoughts are frequently associated with an underlying mental health condition that is treatable. If this is the case, you might need to contact a psychiatrist or another mental health professional who focuses on the diagnosis and treatment of mental illnesses.

Physical health conditions: Suicidal thoughts may occasionally be related to an underlying physical health issue. To ascertain whether this is the case, you may require blood testing and other procedures.

Drug and alcohol misuse: Many people's suicidal thoughts and actual suicides are influenced by alcohol or drugs. If you regularly binge drink or use drugs, or if you find it difficult to reduce or stop using them on your own, your doctor will want to know. Many persons who experience suicidal thoughts require medical attention in order to stop abusing drugs or alcohol and lessen their suicidal thoughts.

Medications: Some people may experience suicidal thoughts when using certain prescription or over-the-counter medications. Inform your doctor about any medications you take so they can check to see if they may be contributing to your suicidal thoughts.

Adolescents and Children: Youngsters who are feeling suicidal should typically be evaluated by a psychologist or psychiatrist who has experience identifying and treating young patients with mental health issues. The doctor will also want to acquire a complete picture of what's going on from a variety of sources, including the patient's family members, friends, school records, and previous medical or psychiatric evaluations.

2.5 Situation in Ireland

According to the data, the suicide rate per 100,000 people was 162.17 in 1998. After that, there was a sharp fall until 2005, when it was 121.11 per 100,000 people. Following that, the suicide rate changed at randomly over next few years. According to the most recent data, the suicide rate per 100,000 people in 2015 was 126.1.

Legal determination of the cause of death(2011 - CSO)**:** Over the five years from 2007 to 2011, the average yearly death rate was over 28,000. The deceased was being treated by a doctor, for example, therefore in many situations the reason of death is known. As the reason of death was typically an illness or ailment the deceased suffered, the doctor can complete the Medical Certificate of the Cause of Death in these situations rather easily.

However, the cause of death is not immediately apparent in 20% of all instances (5,000 to 6,000 cases annually), and the matter is then sent to a coroner. The coroner is required to report and investigate deaths that are sudden, inexplicable, violent, and unnatural. The Coroner is an independent official who is accountable under the law for the medico-legal investigation.

Year	Male	Female	Total
2000	395	91	486

2001	429	90	519
2002	387	91	478
2003	386	111	497
2004	406	87	493
2005	382	99	481
2006	379	81	460
2007	362	96	458
2008	386	120	506
2009	443	109	552
2010	405	90	495
2011	458	96	554

Table 2.1: Deaths by suicide classified by year of occurrence and sex 2000-2011

In table 2.1, Ireland's total suicide is given by the CSO statistical release (2014), You can see there is raise in the overall suicide rate in recent years.

Ireland's Education and Training Plan: The HSE National Office for Suicide Prevention (NOSP) is given a broad framework to assist in the coordination, quality assurance, monitoring, and evaluation of the education and training measures defined in the plan by the Education and Training Plan 2021–2022 (version 4). With the help of this study, government agencies, financed organizations, the HSE, community organizations, groups, and people will be better able to recognize and assist those who are at risk of suicide and self-harm. The plan is centred on five goals, namely

i. the provision of a range of uniform training courses for members of the public, community caregivers, professionals, and volunteers.
ii. Offer training and teaching programs on suicide prevention and mitigation that complement the work of front-line health and social care professionals.
iii. Develop a National Quality Assurance Framework to ensure a uniform and standardized approach to the delivery of education and training.
iv. In accordance with Connecting for Life, assess and monitor the value of suicide prevention training and education.

v. By creating the proper processes, structures, and roles at the national and CHO Area levels, you can oversee the coordination and execution of the education and training plan.

To "provide a consistent and uniform approach to the provision of education and training through the development of a National Quality Assurance Framework (QAF)," according to Objective 3 of the Education and Training Plan. The resulting Quality Assurance Framework (QAF) for the National Education and Training Plan is a dynamic document that will be regularly evaluated and updated to reflect new advancements and best practices in suicide prevention training.

2.6 Models for prediction in Dash App

Time Series Forecasting with ARIMA, SARIMA and SARIMAX

The abbreviation for the ARIMA model is "Auto-Regressive Integrated Moving Average," and for the purposes of easy understanding, we shall separate it into AR, I, and MA (Brendan Artley, 2022).

AR stands for autoregressive component (p). The number of lagged series we employ is determined by the p parameter, which represents the autoregressive part of the ARIMA model as AR(p).

$$y_t = c + \sum_{n=1}^{p} \alpha_n y_{t-n} + \epsilon_t$$

Formula of AR

AR (0) - White Noise

If the p parameter is set to zero (AR (0)), there are no autoregressive terms. White noise is all that this time series is. Each data point is taken as a sample from a distribution with mean 0, variance 2, and standard deviation 0. This generates an unpredictable series of random numbers. This is quite helpful as a null hypothesis since it prevents our analysis from accepting false-positive patterns.

Oscillations and Random Walks in AR (1)

With the p parameter set to 1, we are multiplying the prior timestamp by a factor, then including white noise. A random walk results from a multiplier of 1, while white noise is produced by a multiplier of 0.

The time series will display mean reversion if the multiplier is in the range of $0 < \alpha_1 < 1$. This indicates that after regressing from the mean, the values tend to float around 0 and return to it.

Higher-order terms, AR(p)

The only way to increase the p parameter further is to add more timestamps that have been multiplied on their own. Although we can go back as long as we wish, it is increasingly likely that we should employ more factors, like the moving average (MA(q)), as we go further back.

ARMA and ARIMA Models

The AR (Autoregressive) and MA (Moving Average) components alone make up the ARMA and ARIMA architectures.

ARMA: The ARMA model consists of a constant plus the sum of the AR and MA lags and their multipliers, plus white noise. This equation serves as the foundation for all subsequent models and as a framework for numerous forecasting models in various fields.

$$Y_t = \beta_2 + \omega_1\varepsilon_{t-1} + \omega_2\varepsilon_{t-2} + \ldots + \omega_q\varepsilon_{t-q} + \varepsilon_t$$

The ARIMA model is an ARMA model, however the model we describe using I includes a preprocessing step (d). The difference order, I(d), indicates how many transformations are required to make the data steady. An ARMA model on the differenced time series is what an ARIMA model is thus.

Models SARIMA, ARIMAX, and SARIMAX: Although the ARIMA model is excellent, adding seasonality and exogenous variables can have a significant impact. We must employ a different model because the ARIMA model presumes that the time series is stationary.

$$y_t = c + \sum_{n=1}^{p} \alpha_n y_{t-n} + \sum_{n=1}^{q} \theta_n \epsilon_{t-n} + \sum_{n=1}^{P} \phi_n y_{t-sn} + \sum_{n=1}^{Q} \eta_n \epsilon_{t-sn} + \epsilon_t$$

Except for the additional set of autoregressive and moving average components, this model and the ARIMA model are quite similar. The frequency of seasonality (ex. 12 monthly, 24 hourly) cancels out the extra lags. SARIMA models allow for both seasonal frequency and non-seasonal frequency differences in data. Automatic parameter search frameworks like pmdarima (Documentation) can make it simpler to determine which parameters are ideal.

$$d_t = c + \sum_{n=1}^{p} \alpha_n d_{t-n} + \sum_{n=1}^{q} \theta_n \epsilon_{t-n} + \sum_{n=1}^{r} \beta_n x_{n_t} + \sum_{n=1}^{P} \phi_n d_{t-sn} + \sum_{n=1}^{Q} \eta_n \epsilon_{t-sn} + \epsilon_t$$

The SARIMAX model is seen in fig 4.7. Exogenous variables are used in this model, or in other words, external data is used in our forecast. Examples of exogenous variables in the real world include the price of gold, oil, the outside temperature, and the exchange rate. It's interesting to consider that all external variables are still de facto implicitly modeled in the forecast from the historical model. However, if we incorporate external data, the model will react to its impact much more quickly than if we rely just on the influence of lagging components.

FB Prophet

An open-source library called Facebook Prophet forecasts time series data. It assists both individuals and companies in analyzing market values and forecasting the future. It puts into practice a method for predicting time series data that is based on an additive model where non-linear trends are fit with yearly, monthly, and daily seasonality, as well as holiday impacts. It functions best with historical data from multiple seasons and time series with seasonal impacts. Prophet is a method for predicting time series data that uses an additive model to fit non-linear trends with seasonality that occurs annually, monthly, daily, and on weekends as well as during holidays. Strongly seasonal time series and multiple seasons of historical data are ideal for it. Prophet typically manages outliers well and is robust to missing data and changes in the trend.

Trend: A pattern can be seen in the data. The time series data's non-periodic variations are modeled. The dataset's long-term movement is depicted by a trend. A trend may be constant, uphill (uptrend), or downhill (downtrend) (horizontal). Trends typically emerge for a while before disappearing.

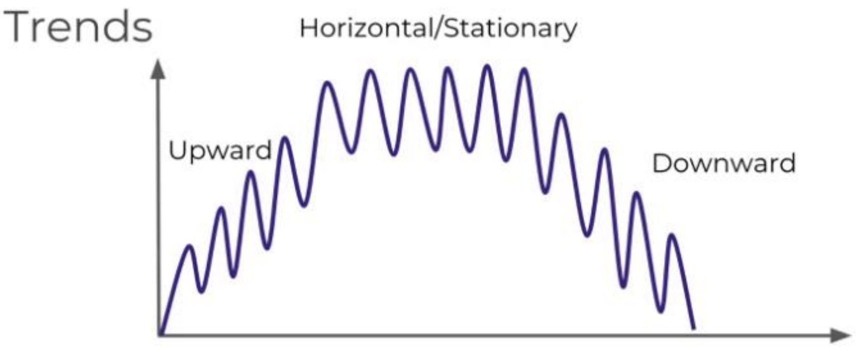


Fig 2.2 Different ***Seasonality in time series data***

Seasonality: This happens due to the daily, weekly, and yearly changes in the data.

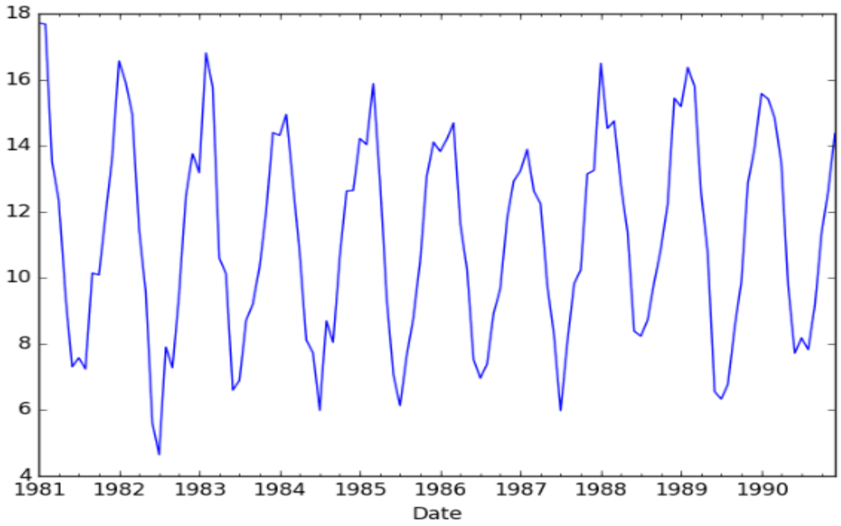


Fig 2.3 Holiday effect in time series data

Holiday effect: In a time-series dataset, these are the recurring days and events. It is concerned with the recurrence of well-known holidays like Christmas and others.

Benefits of using Facebook Prophet

The advantages of using Facebook Prophet for time series modelling are as follows.

The prophet model is utilized in numerous Facebook applications to generate accurate forecasts for planning and goal setting. In most instances, it is discovered that it outperforms every other strategy. So that you can receive forecasts in only a few seconds, using the fitted models in Stan (Documentation).

Get an accurate forecast from muddled data without any manual work. Prophet can withstand outliers, missing data, and significant time series changes. Additionally, data points that differ from the main dataset observations are removed. It can manage the impacts of seasonality and holidays. It manages the spikes in the dataset and takes them into account while training the model.

Users of the Prophet method have a lot of options for modifying and adjusting forecasts. By incorporating your subject knowledge, you can employ human-interpretable features to enhance your forecast.

Both R and Python have a prophet technique, but they both use the identical Stan fitting code.

Custom AR

Custom AR is used because it is more flexible in terms of customization.

$$\boxed{\hat{Y} = \sum_{i=1}^{n} Wi\, Xi}$$

Where X_i is the value in the i^{th} previous year and W_i is the weight or coefficient of regression. Custom AR is manually written, and it is the foundation on which other models are created. The Custom Auto regression model is created because this study needed a transparent working mechanism to see how these forecasting algorithms work. Linear Regression Algorithm has been used in fitting the model. A function named train_and_forecast is used in the data_modelling.py file to create the models and run the forecast. AR_forecast function runs the previously created train_and_forecast function and returns error and prediction from it.

Chapter 3

Exploration of Data

3.1 Data Collection

Dataset has been directly downloaded from the Kaggle website. As you can see from the fig 3.1 data has been directly downloaded and stored into an encrypted HP laptop hard drive. Data has been loaded to Jupyter Notebook inside anaconda environment for initial analysis(EDA). The main objective of this research is to bring underlying issues with suicides happening around the world, neither data scrapping was required nor important to the main objectives of the study. The best part of using Kaggle Dataset was that it did not require any extra effort to write scripts on how to bring the dataset from APIs like twitter API or YouTube API. Moreover, this research is more of visualisation and prediction oriented where web scrapping or API data collection are less important.

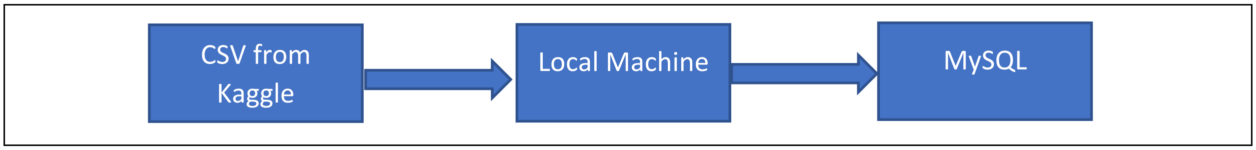


Fig 3.1 Data Collection from internet to SQL

Structure of data needed for SQL loading is compared with database schema before uploading the collected data to the DB. Types such as int, varchar, float etc. are checked on the final output.csv file making sure structure DB and collected and cleaned dataset matches the table structure.

3.2 Data Preparation

Data imported to the Jupyter notebook file has been analyzed using EDA and statistical analysis techniques. The first stage is to check for correct data types. Data in the form of data frame will be analyzed with panda's package. Describe function gives an overview to the data. After fixing the outliers, duplicate rows have been checked and treated with care. If the entire row having most of the information repeated from the previous row, then that have been removed. Outliers can be checked using different methods such as box

plots or statistical methods. I have used some statistical approach to pull out the extreme values. DB scan has been applied to handle the outliers.

Decompose the Data(Vera Shao, 2020): In case of a broad upward trend without any discernible seasonal or cyclical patterns. The data must then be broken down to see more of the complexity that lies behind the linear visualization. We may divide the data into four parts with the help of the seasonal decompose Python function from the statsmodels package.

Any time series data should be examined for trends that could have an impact on the outcomes and could guide the selection of the forecasting model. Several typical time series data patterns include:

I. Level: The average value in the series
II. Trend: Increases, decreases, or stays the same over time
III. Seasonal or Periodic Pattern: Pattern repeats periodically over time
IV. Cyclical Pattern: Pattern that increases and decreases but usually related to non-seasonal activity, like business cycles
V. Random or Irregular Variations: Increases and decreases that don't have any apparent pattern

3.3 Description of the data

The dimension of the first dataset is 27820 observations from 1998 to 2015. This data contains duplicated records, outliers, missing data as well as wrong observations. After cleaning the dataset, the data frame size became 15110. 12,710 records were removed during the cleaning process. Final dataset called output.csv contain 26 features. Only relevant columns are used for visualizing the dashboard. All of the charts in EDA were created using data from thirty-nine nations.

Master dataset and second dataset are combined and cleaned using DB scan to get the final dataset. Data contains all the required variables such as categorical, numeric and object. Continent and country code are added from a third dataset called countryContinent.csv. All the files required for the project are stored in the asset folder of python dashboard repository.

Sl No.	Column Name	Description
1	country	Name of the country
2	year	Year in which the suicides happened
3	sex	Gender of the suicide case
4	age	Age of the people died
5	suicides	Number of total suicides in a country
6	population	Population in the country by year
7	sucid_in_hundredk	Suicides happened in hundred thousand.
8	country-year	Country and year together as a category
9	yearly_gdp	yearly gdp of each country
10	gdp_per_capita	gdp per capita of each country
11	generation	Generation categories of the suicides
12	suicide%	Suicide percentage of each year record
13	internetusers	Yearly Internet users in each country
14	expenses	Yearly Expenses in each country
15	employeecompensation	Yearly Employee ation in each country
16	unemployment	Yearly Unemployment figure in each country
17	physician_price	Yearly Physician price in each country
18	laborforcetotal	Yearly Laborforforce total in each country
19	lifeexpectancy	Yearly Lifeexpectancy in each country
20	mobilesubscriptions	Yearly Mobilesubscriptions in each country
21	refugees	Yearly Refugees ugees in each country
22	selfemployed	Yearly Selfemployed in each country
23	electricityacess	Yearly Electricityacess of people in each country
24	continent	Continenent in which the country belongs to
25	country_code	Countrycode of each country
26	mobilesubscription	Yearly Mobilesu ption in each country

Table 3.1: Column names and description.

Python Dashboard in Plotly

Python dash and streamlit have emerged as the most capable frameworks for web-based visualisation projects in recent years. This project provides both static and dynamic visualisations. Before the real web dashboard app, individual static graphs are produced to obtain insight from the data. A final dashboard app with dynamic visualisations will be constructed when the initial static models are completed in Jupyter Notebook. It's always wonderful to see how we can make models and interpret them. But it is also important to note, recently there are number of concerns about how well we can make modifications to the existing model and maintain them. So, our model must work dynamically and make predictions based on the available data. In recent years programmers used use VueJS or web-based languages for making dashboards, we now have most advanced packaged like has made these process easier and more efficient. I am going to use some of the python packages like plotly to make interactive dashboard and make models that can make great predictions. The following visualisations are done using data 1985 to 2015

Suicide per hundred thousand around the world

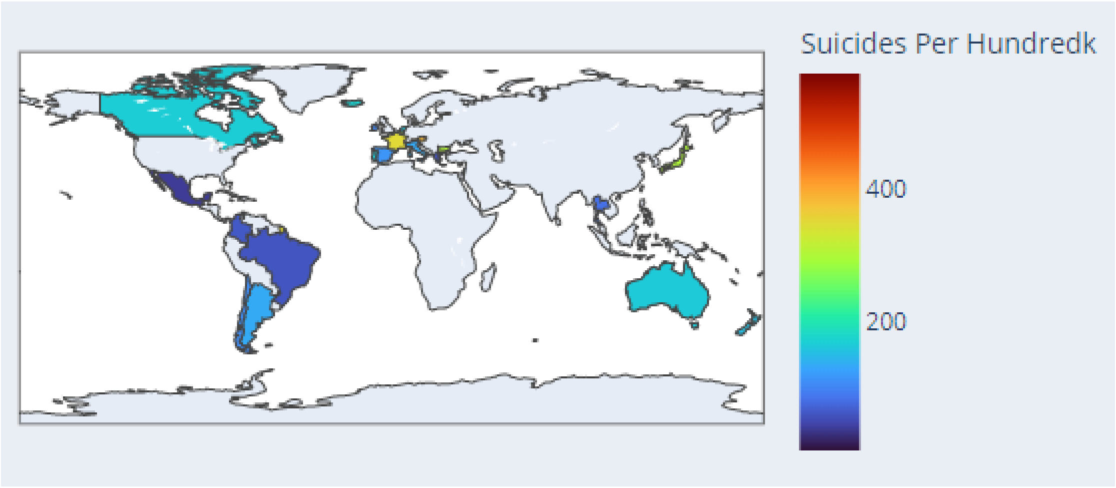


Fig. 3.2: Suicide per hundred thousand around the world - Timeline in plotly

Plotly is used to depict the global suicide rate in fig 4.3. Visitors can see information based on the year on an animation frame page. The rate of suicides per 100,000 is shown by colored zones.

Suicide per hundred thousand in Male and Female

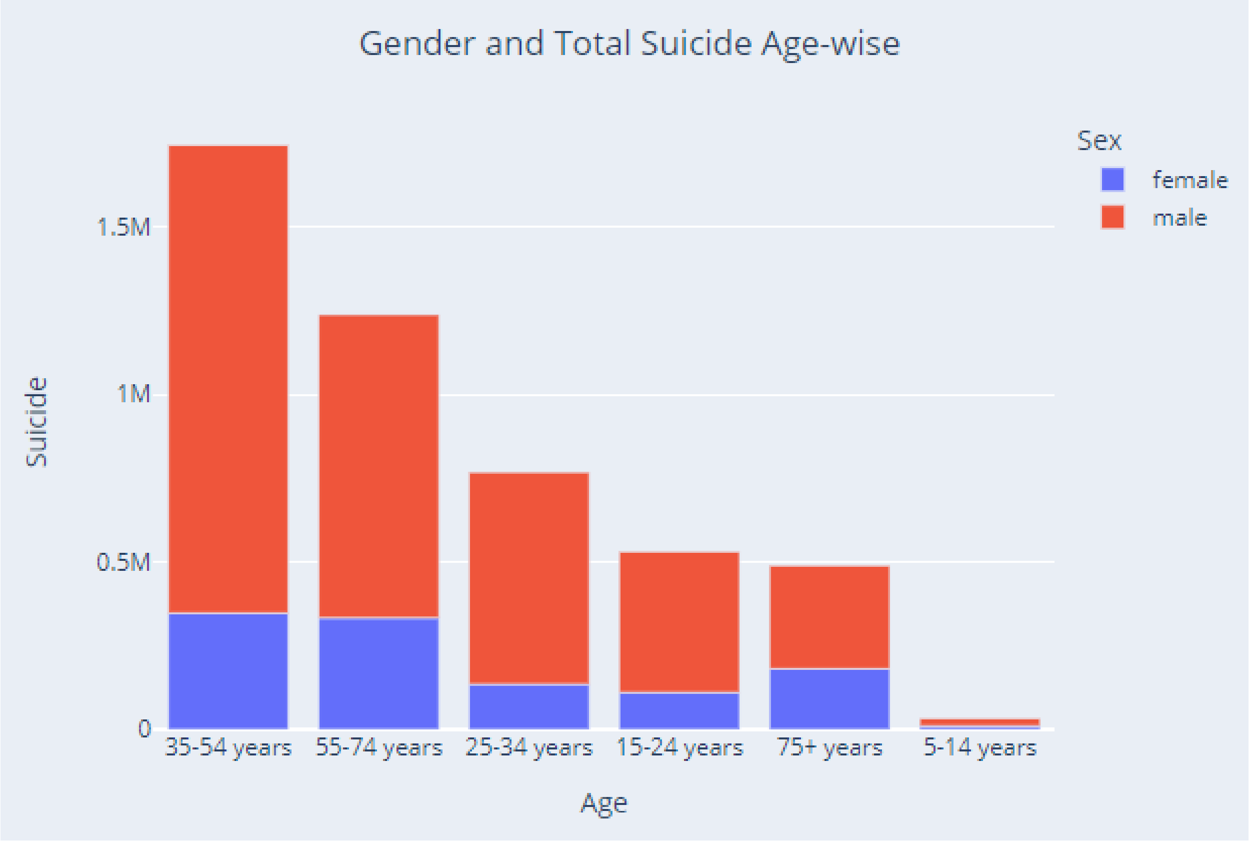


Fig. 3.3: Suicide among males and females in different age groups in different countries

As per fig 3.3, Most suicides are happening between the age of 35 and 54. And out of the majority are Males. In all the age groups females are less affected groups. Also, we can see from the age of five to fourteen children are less likely to commit suicide.

Suicide per hundred thousand Vs GDP Per capita

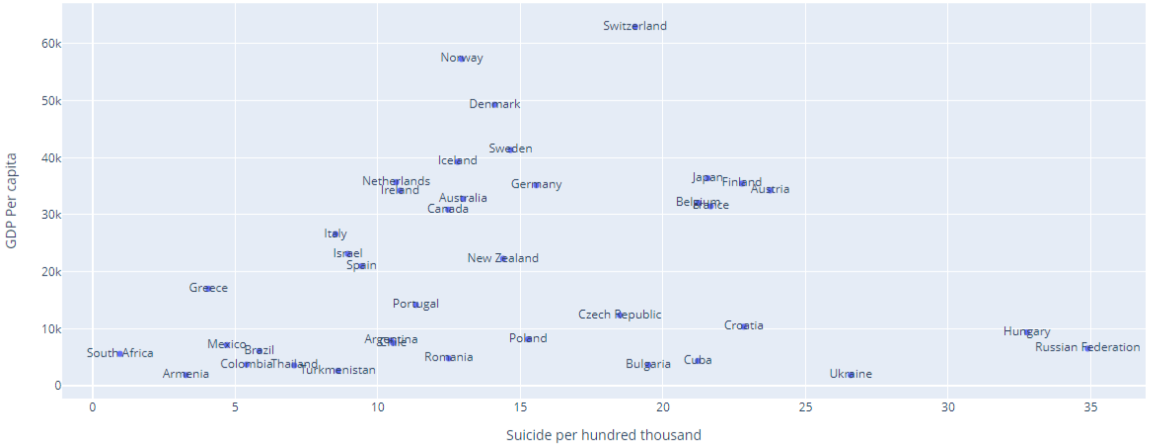


Fig. 3.4: Suicide per hundred thousand around the world in different countries

As seen in the fig 4.5, nations such as Russia, Ukraine, and Hungary have some form of relationship in terms of the number of suicides per 100,000 people. According to a BBC News article, the causes of suicide in Ukraine are the consequences of Russia-Ukraine hostilities and the ongoing war. Ukraine is undoubtedly one of the most afflicted countries in terms of suicide, according to our statistics. As per the figure 3.4, Moderate GDP Per capita has a higher suicide rate shows there is a relation between both variables.

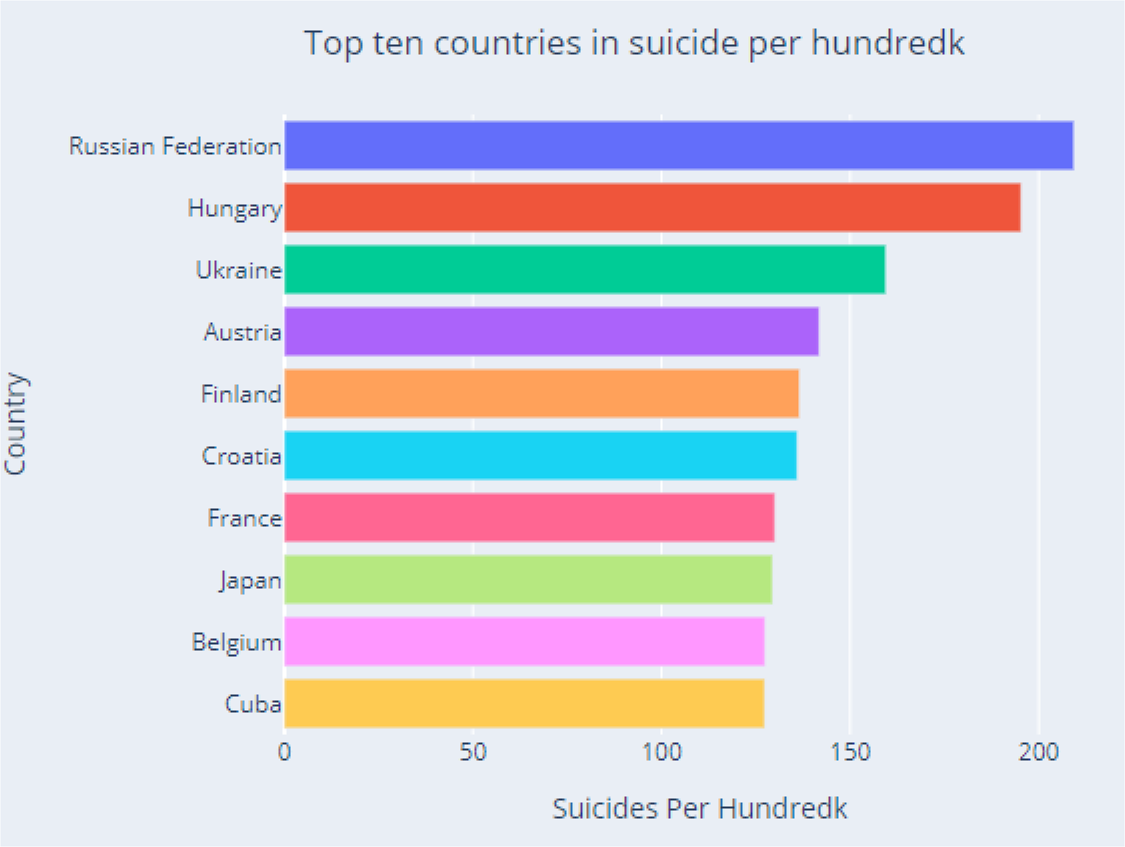


***Fig. 3.5:** Top ten countries with the highest suicide rates*

The extent of suicides in various nations is seen in Figure 3.5. It is apparent that the Russian Federation has the highest suicide rate of all the countries studied. According to earlier research by Bellman and Namdev (2022), Russia has a considerable problem with suicidal behavior among men, and their drinking habits have a substantial impact on their decision to commit suicide when compared to other nations.

Population, Suicide, Suicide in Hundred Thousand Vs Gender and Age

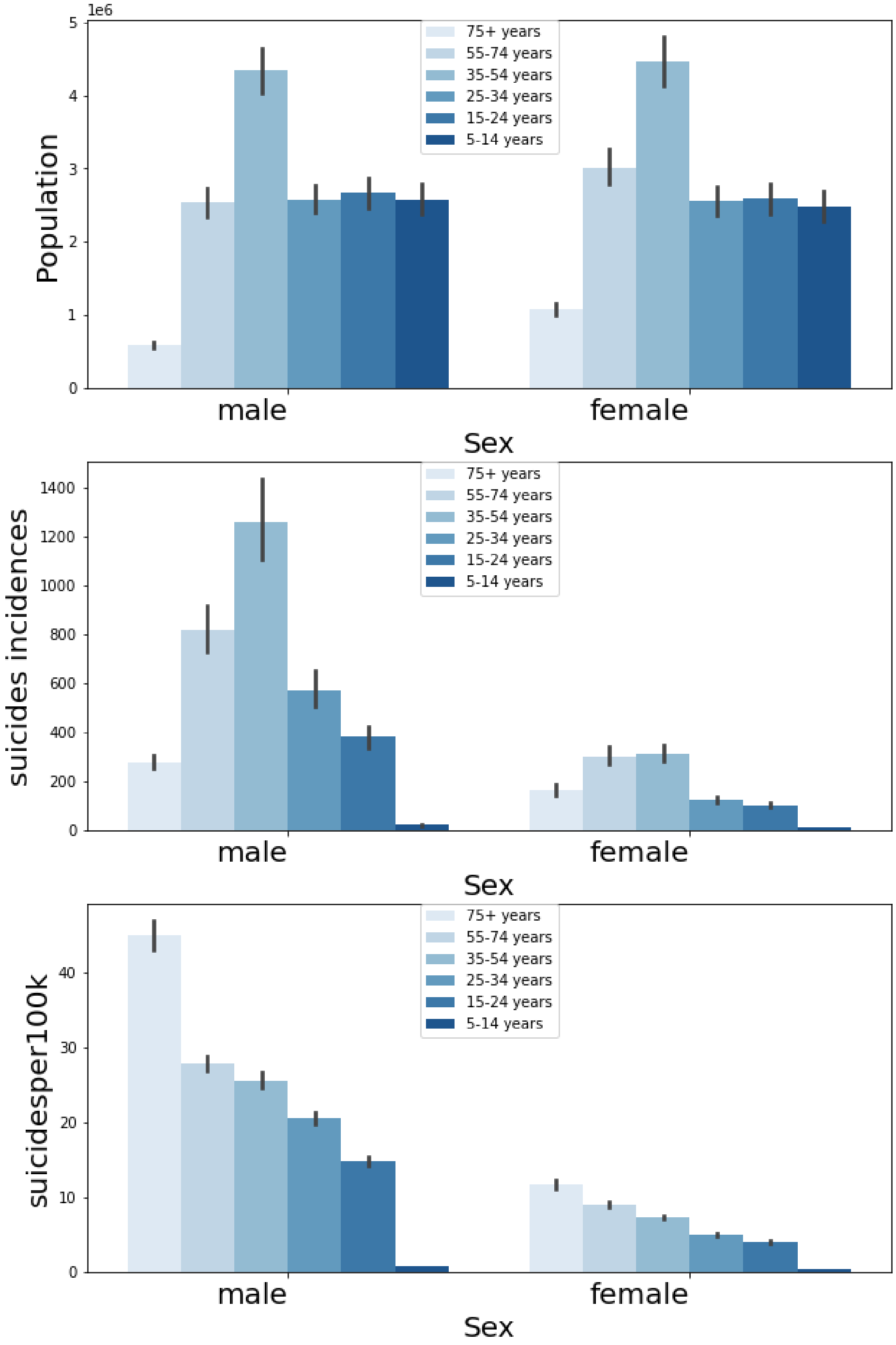


Fig. 3.6: Suicide, Suicide per hundred thousand, Population in different genders and age in all countries

As seen in fig. 4.7, Females outnumber men in terms of population. In relation to total suicides among men and women, the majority occurred between the ages of thirty-five and fifty-four. It is undeniable that the male population has a higher suicide rate per hundred thousand. Furthermore, the majority of those who died were above the age of seventy-five with respect to suicide per hundred thousand.

Population, Suicide, Suicide in Hundred Thousand Vs Gender and Age

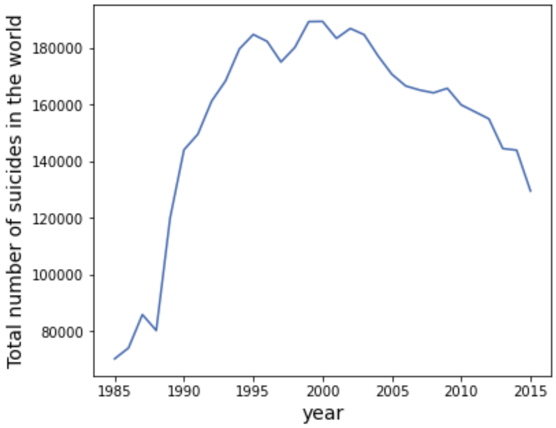


Fig. 3.7: Total suicides in the world in each year distribution

Fig 3.7 shows that there is a quick rise in the suicide rate from the year 1990. After that, it continued increasing until the next ten years. From 2000 it started to decline. The global decrease has many causes(The Economist, 2018), but three groups of individuals stand out in particular. Young ladies in China and India are one example. Men kill themselves more frequently than women over the world, and older people are more likely to do so than young ones. But young women have an abnormally high suicide rate in China and India. That's becoming less and less true. Since the middle of the 1990s, the rate among young Chinese women has decreased by 90%. Russian guys in their middle age are another category. Alcoholism and suicide rates among them skyrocketed after the fall of the Soviet Union. Now, both have diminished. The elderly worldwide comprise a third category. Although the suicide rate among the elderly has decreased more quickly than that of other categories since 2000, it still remains higher than that of the general population on average.

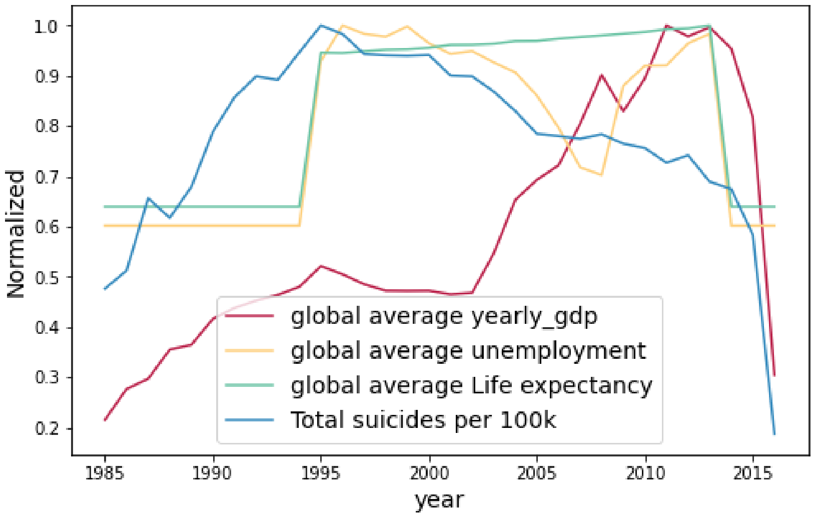


Fig. 3.8: Global Distribution of different features

According to fig 3.8 above, the number of suicides per 100,000 climbed from 1985 to 1995, then rapidly fell. At the same time, global GDP per capita climbed gradually from 1985 to 1995, remained stable until 2003, then increased abruptly until 2014, before falling precipitously in 2015 (UN, 2015). Life expectancy and unemployment have been influenced by missing data and imputation, making the lines appear irrelevant. A huge decline at the end of the graph is due to the incomplete data from 2015.

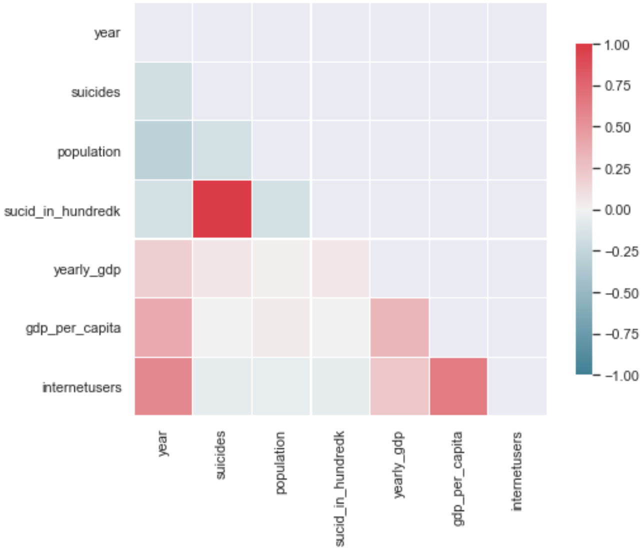


Fig. 3.9: correlation matrix of suicide dataset

Normally, checking for features in the correlation matrix is necessary to see whether there are any features with strong correlation. We usually eliminate such variables from the dataset since they might cause the model to overfit. As seen in fig. 4.10, there is a significant correlation between suicide and suicide per hundred thousand. As a result, before modeling, the suicides feature was deleted from the Data frame.

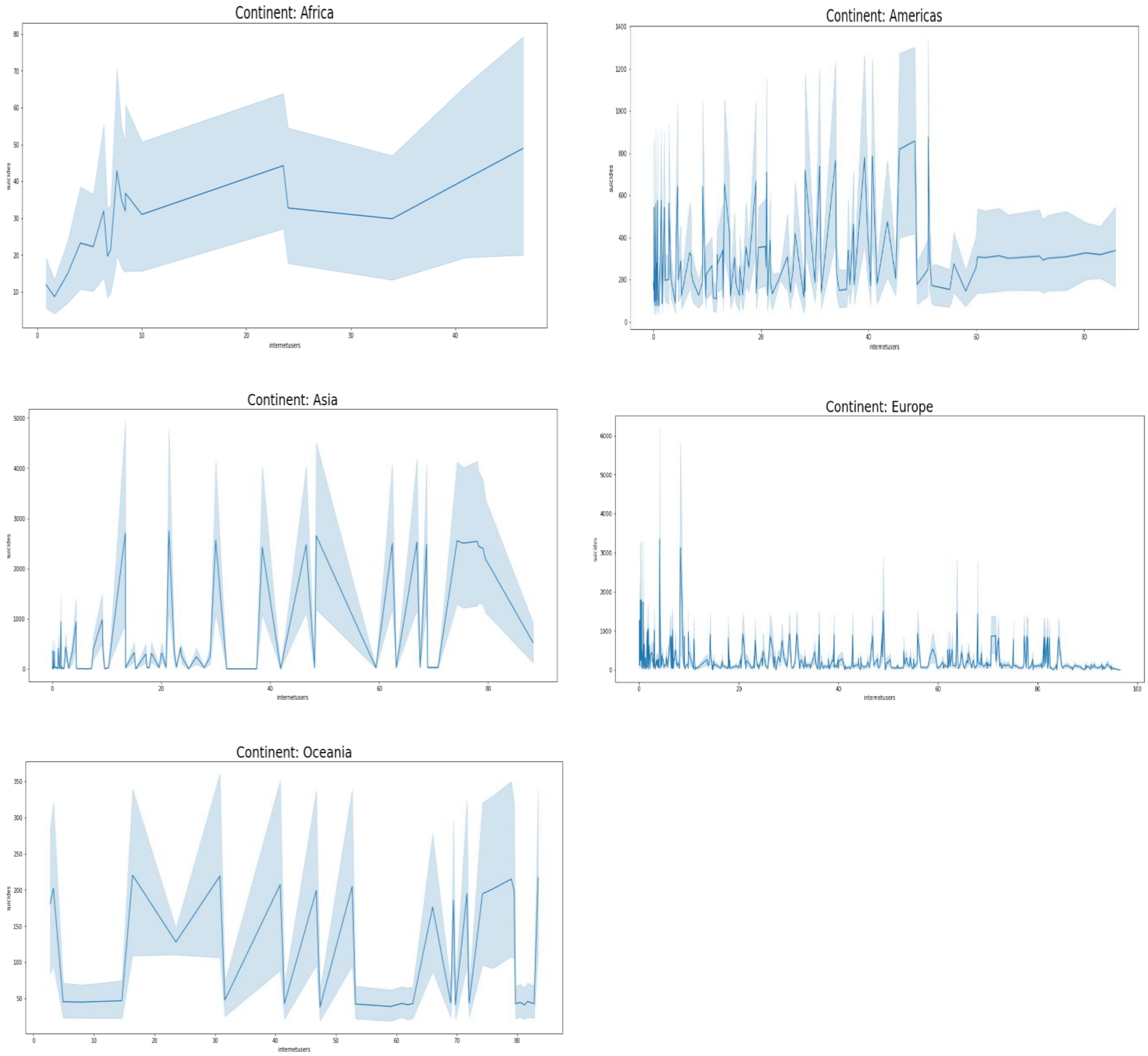
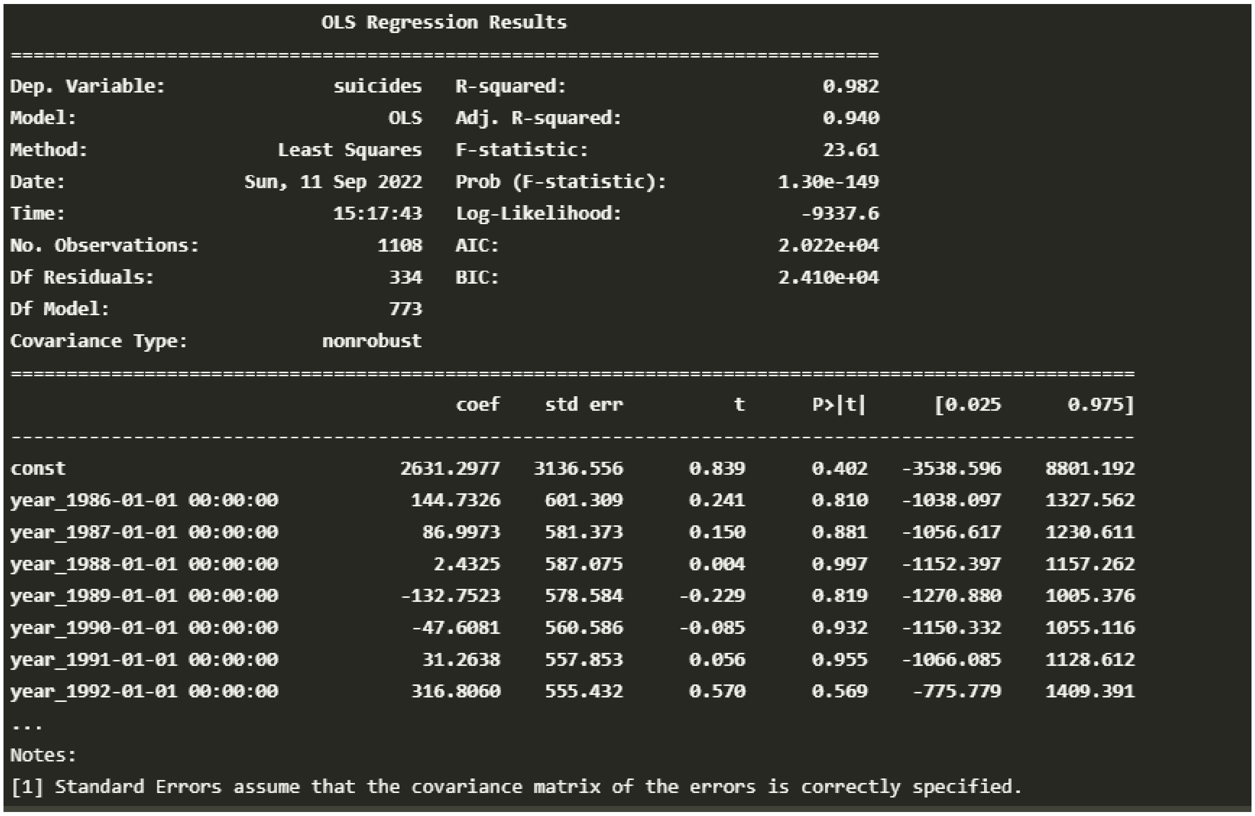


Fig 3.10 Suicide Trend Vs Internet usage

As the number of internet users across Africa went up, so did the incidence of suicides. There is an upward trend in total suicides concerning internet usage. In America, suicide rates fluctuated erratically up to a certain period, after which no change in suicide rates was seen. It seems to have a steady line meaning there is a change in suicides from sixty or more internet users. In Asia also as internet users increase the number of suicides also increases with random suicide figures in between. In Europe, the trend seems to be the opposite of Asia. As the number of internet users increases, suicide decreases.



```
                            OLS Regression Results
==============================================================================
Dep. Variable:               suicides   R-squared:                       0.982
Model:                            OLS   Adj. R-squared:                  0.940
Method:                 Least Squares   F-statistic:                     23.61
Date:                Sun, 11 Sep 2022   Prob (F-statistic):          1.30e-149
Time:                        15:17:43   Log-Likelihood:                -9337.6
No. Observations:                1108   AIC:                         2.022e+04
Df Residuals:                     334   BIC:                         2.410e+04
Df Model:                         773
Covariance Type:            nonrobust
=====================================================================================================
                                        coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------------------
const                              2631.2977   3136.556      0.839      0.402   -3538.596    8801.192
year_1986-01-01 00:00:00            144.7326    601.309      0.241      0.810   -1038.097    1327.562
year_1987-01-01 00:00:00             86.9973    581.373      0.150      0.881   -1056.617    1230.611
year_1988-01-01 00:00:00              2.4325    587.075      0.004      0.997   -1152.397    1157.262
year_1989-01-01 00:00:00           -132.7523    578.584     -0.229      0.819   -1270.880    1005.376
year_1990-01-01 00:00:00            -47.6081    560.586     -0.085      0.932   -1150.332    1055.116
year_1991-01-01 00:00:00             31.2638    557.853      0.056      0.955   -1066.085    1128.612
year_1992-01-01 00:00:00            316.8060    555.432      0.570      0.569    -775.779    1409.391
...
Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
```

Fig 3.11 Multi Linear Regression on suicide data

The R-squared of the model, which is 0.982 (98.2%) when we look at the results summary, shows that the model is performing well.

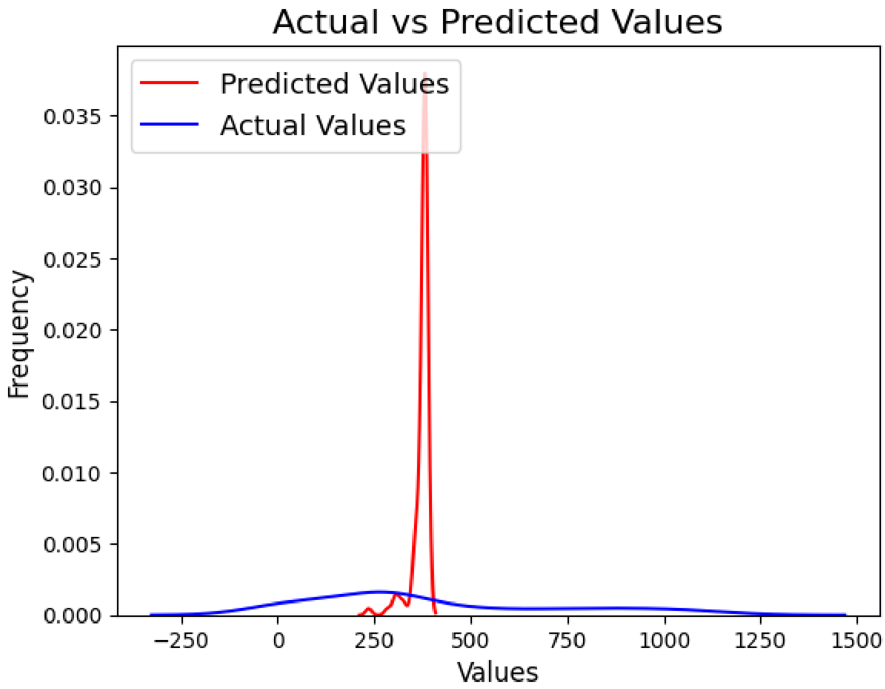


Fig 3.12 Comparison of actual and predicted values from MLR model.

Europe seems to be more aware of technology than Asian Society, they are less affected by the usage of the internet or social media. In the case of Oceania, there is no specific trend found in the data. Suicides with respect to internet users seem to be random in this continent. A Multi Linear regression Model has been created to statistically check the relationship between internet usage and suicides as per fig 3.10. It seems to be not much relation going on between internet usage and suicide as per the testing results.

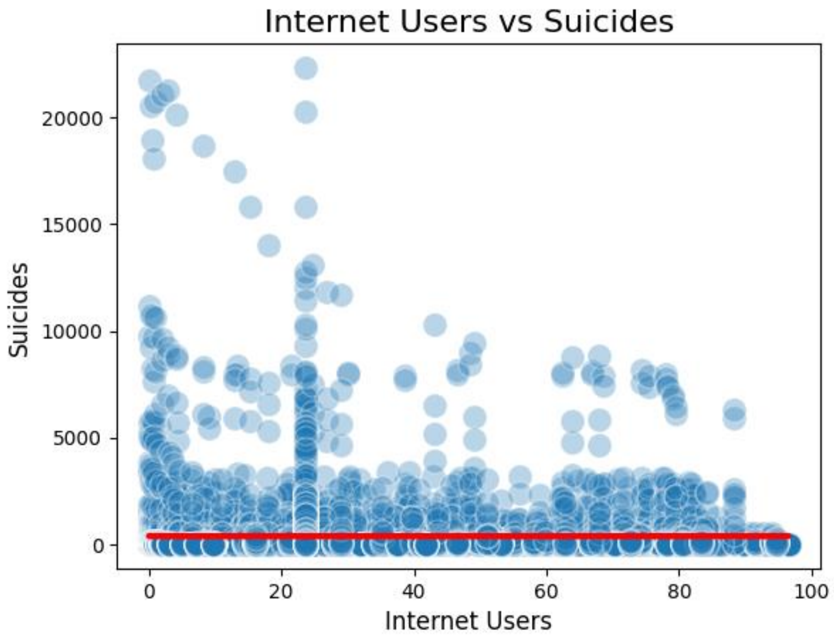
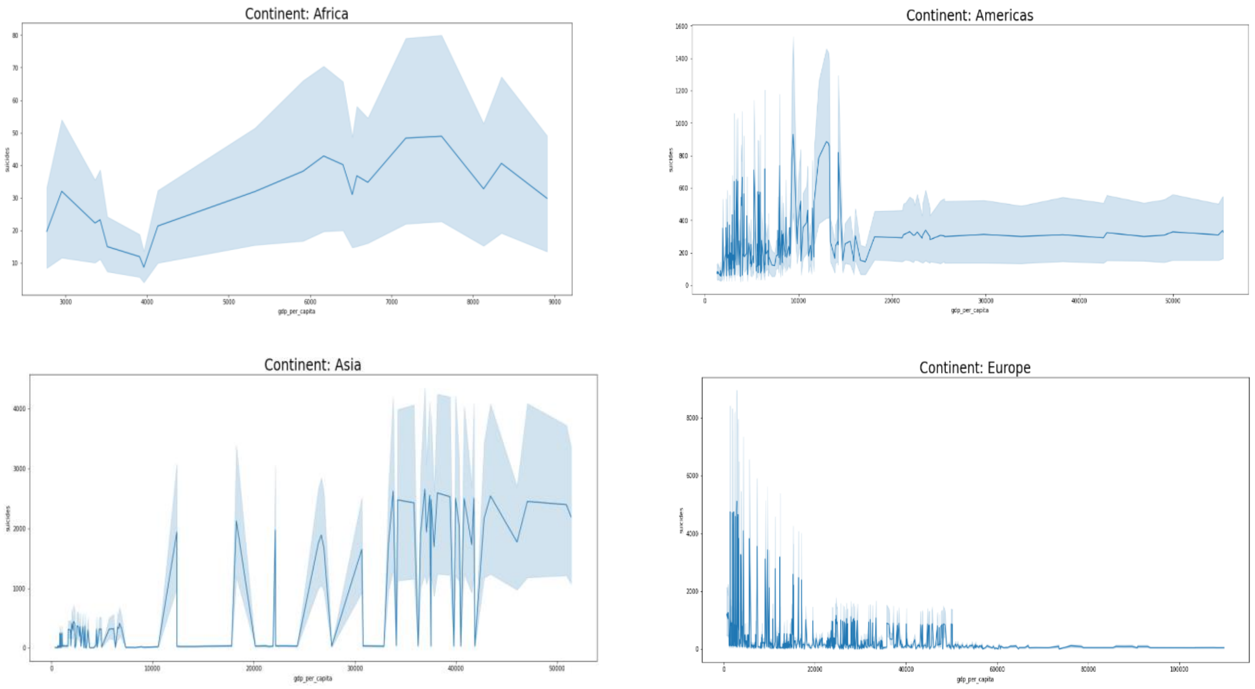


Fig 3.13 Multi Linear regression for suicide and internet users

In terms of GDP Per capita, African continent does not show any trend in number of suicides. African countries are not only struggling with poverty but also with lots of other basic issues like food, water(Khanyi Mlaba, 2022) etc. Other issues Africa faces are mentioned in this article from Africa portal (Steven Gruzd, 2022).



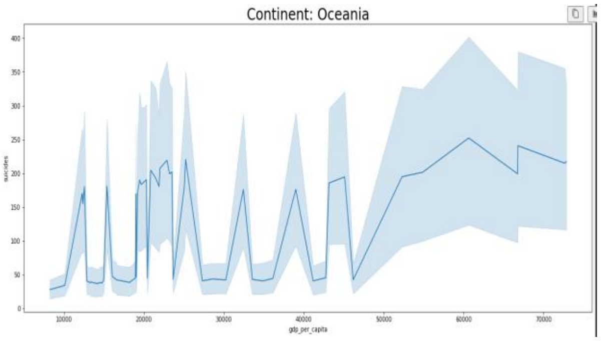


Fig 3.14 Suicide Trend Vs GDP Per capita

In American continent the situation is entirely different. Suicides have a high influence on countries with less GDP per capita. Suicides are more affected on countries having very less GDP per capita. As it increases after a certain level number of suicides seems to stay steady. In America number of suicides is almost same in countries having more than twenty thousand GDP per capita. In Asia suicides have an increasing trend with respect to increase in GDP per capita. It is understood that people living in countries with high GDP per capita in Asia could be facing more issues compared to the less GDP countries. Countries like India millions of farmers in different states living peaceful life doing farming and day labor jobs happier than the people with high income living in cities struggling to maintain the expectation of highly developed rich people (David Hurst, 2021). Situation in Europe is quite different from Asian continent. Number of suicides are more in countries with less than fifty thousand GDP Per capita. As a highly developed society, to live a happy and peaceful life a high GDP Per capita is required for any country. This shows that authorities in health sector must provide more focus on countries with less GDP Per capita to control this situation. Countries like Russia, Ukraine and Hungary are the best example of this struggle that Europe carries since early nineties. Oceania shows a slightly increase in the suicides as the GDP Per capita increases.

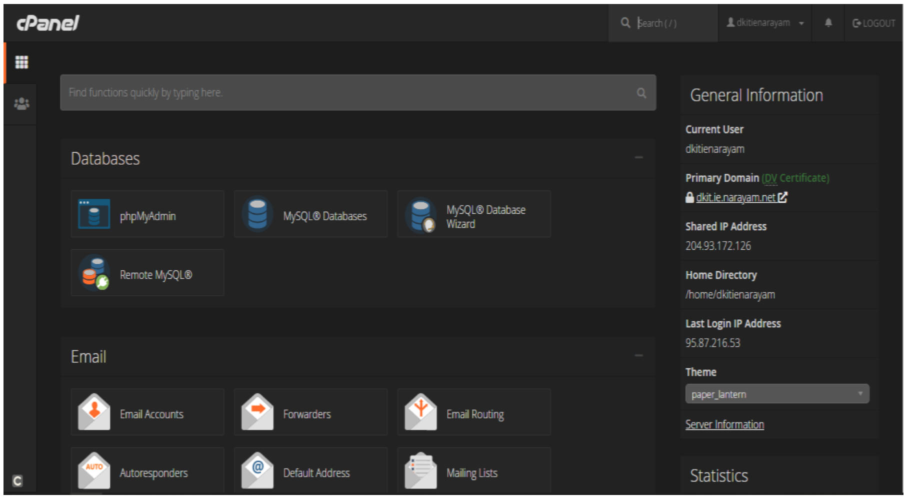
It is interesting to see from above observations how suicides varied in different continents with respect to their GDP per capita. Also, the trends can be different in different continents. This is due to the socio-economic changes affecting the people in each country slightly different in each continent.

Chapter 4

Design and Implementation

4.1 Database design and Migration

For running python packages and preparing the server ready new VPS package has been purchased from Mochahost. pip dash packages have been installed on the server. Initially small bits of code have been written on the local machine inside Jupyter notebook. XAMPP has been also installed on the machine to facilitate Maria DB. Using command prompt csv files have been uploaded to the DKIT database. Using SQL 'LOAD DATA LOCAL INFILE' commands command has been run in order to get he data which is produced by the data_maker.py file inside the suicide_dash_app folder.



Current User
dkitienarayam

Primary Domain (DV Certificate)
dkit.ie.narayam.net

Shared IP Address
204.93.172.126

Home Directory
/home/dkitienarayam

Last Login IP Address
95.87.216.53

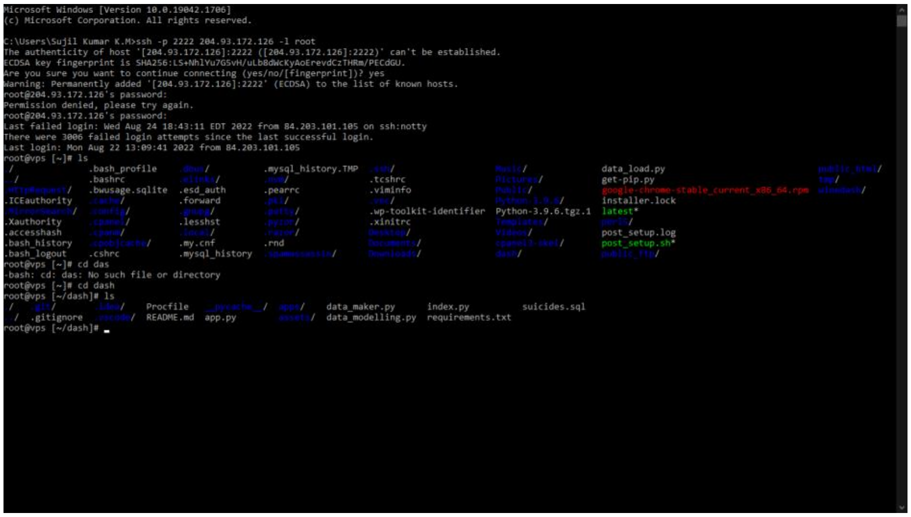
Theme
paper_lantern

Server Information

Fig 4.1 *Cpanel Home Page*

4.2 SSH Access and Putty Configuration

In order to use the SSH access time to time, putty configuration was required. Data architecture skills need to be used to configure or prepare the server. cPanel's official documentation (2021) can be checked to get details on preparation of the server.

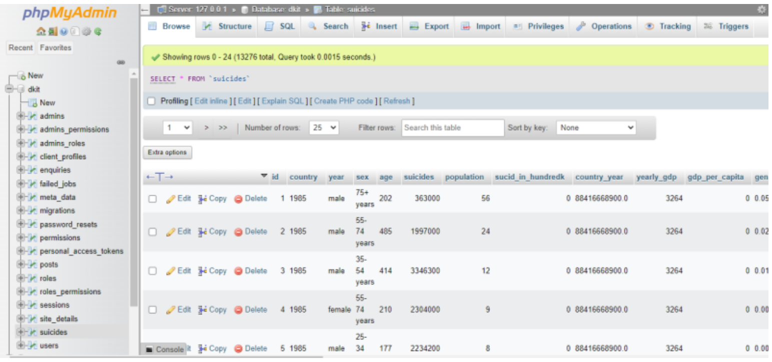


```
Microsoft Windows [Version 10.0.19042.1706]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Sujil Kumar K.M>ssh -p 2222 204.93.172.126 -l root
The authenticity of host '[204.93.172.126]:2222 ([204.93.172.126]:2222)' can't be established.
ECDSA key fingerprint is SHA256:LS+NhlYu7GSvH/uLbBdWcKyAoErevdCzTHRm/PECdGU.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[204.93.172.126]:2222' (ECDSA) to the list of known hosts.
root@204.93.172.126's password:
Permission denied, please try again.
root@204.93.172.126's password:
Last failed login: Wed Aug 24 18:43:11 EDT 2022 from 84.203.101.105 on ssh:notty
There were 3006 failed login attempts since the last successful login.
Last login: Mon Aug 22 13:09:41 2022 from 84.203.101.105
root@vps [~]# ls
./               .bash_profile    .dbus/         .mysql_history.TMP  .ssh/                      Music/                 data_load.py                           public_html/
../              .bashrc          .elinks/       .npm/               .tcshrc                    Pictures/              get-pip.py                             tmp/
.HttpRequest/    .bwusage.sqlite  .esd_auth      .pearrc             .viminfo                   Public/                google-chrome-stable_current_x86_64.rpm  [illegible]/
.ICEauthority    .cache/          .forward       .pki/               .vnc/                      Python-3.9.6/          installer.lock
.MirrorSearch/   .config/         .gnupg/        .putty/             .wp-toolkit-identifier     Python-3.9.6.tgz.1     latest*
.Xauthority      .cpanel/         .lesshst       .pyzor/             .xinitrc                   Templates/             perl5/
.accesshash      .cpanm/          .local/        .razor/             Desktop/                   Videos/                post_setup.log
.bash_history    .cpobjcache/     .my.cnf        .rnd                Documents/                 cpanel3-skel/          post_setup.sh*
.bash_logout     .cshrc           .mysql_history .spamassassin/      Downloads/                 dash/                  public_ftp/
root@vps [~]# cd das
-bash: cd: das: No such file or directory
root@vps [~]# cd dash
root@vps [~/dash]# ls
./   .git/        .idea/    Procfile   __pycache__/  apps/    data_maker.py      index.py          suicides.sql
../  .gitignore  .vscode/  README.md  app.py        assets/  data_modelling.py  requirements.txt
root@vps [~/dash]# _
```

Fig 4.2 SSH access to the server and files in the dash app

4.3 DB Schema and table creation in Laravel

After creating the database, database design is done from Laravel artisan console. The required columns and table names were created as migration files in the Laravel database directory. The appropriate datatype must be defined for each specific variable. Tables required for both the python dash app as well as Laravel backend for admin panel has been defined in one go. "php artisan migrate” command has been used to run the migration. The database section also has database seeders used to inject predefined data like admin names and passwords for super admin access. “php artisan db:seed has been used to run the seed.



✓ Showing rows 0 - 24 (13276 total, Query took 0.0015 seconds.)

```
SELECT * FROM `suicides`
```

1 > >> | Number of rows: 25 Filter rows: Search this table Sort by key: None

	id	country	year	sex	age	suicides	population	sucid_in_hundredk	country_year	yearly_gdp	gdp_per_capita	gen
□ Edit Copy Delete	1	1985	male	75+ years	202	363000	56	0	88416668900.0	3264	0	0.05
□ Edit Copy Delete	2	1985	male	55-74 years	485	1997000	24	0	88416668900.0	3264	0	0.02
□ Edit Copy Delete	3	1985	male	35-54 years	414	3346300	12	0	88416668900.0	3264	0	0.01
□ Edit Copy Delete	4	1985	female	55-74 years	210	2304000	9	0	88416668900.0	3264	0	0.00
it Copy Delete	5	1985	male	25-34	177	2234200	8	0	88416668900.0	3264	0	0.00

Fig 4.3 Mysql DB Loaded with data

4.5 Contribution of PHP Framework

In earlier days, people used tell that PHP is slow and unsafe framework, as new versions of PHP have evolved, new frameworks like Code Igniter, Laravel were came into action with more sophisticated technical capabilities keeping the application faster, smarter, efficient, and easy to use. Laravel uses MVC Architecture (Model, View, and Controller). It is the most useful any developer can easily rely on in this era (2022). Laravel uses power DB management system called eloquent ORM (Object relational Mapper). Also, Artisan console provides easy access to its boilerplate codes by just typing few commands.

4.6 Modeling and Forecasting

This project has various objectives, including creating a dashboard for visualizations, forecasting using Machine Learning Models, and creating an Admin Panel Portal for updating new suicides etc. Working with a time series model and projecting future values would be the most intriguing and challenging aspect of this endeavor.

Core technology Core technology
Stage: **Modeling process** Programming Language:
Python IDE: **Jupyter Notebook**
1 **sklearn**. For pipeline, splitting dataset and create SVM model. 2 **matplitlib**. To plot graphs for evaluation. 3 **pandas** and **numpy**. For data manipulation. 4 **mysql_connection**. – To store and retrieve data to the dash application.

Table 4.1: Core technology for modelling stage.

A decision tree classifier was employed to categorize the risk and non-risk groups while dealing with numerical and categorical information. In the study of heart failure (Aljaaf et al., 2015) seem to have effectively utilized them in their research, which prompted me to extend the concept to suicide analysis. Checking seasonality, trends, and stationarity, as well as assessing prediction tests for identifying the optimal model for prediction, such as AIC and BIC, are all part of this research.

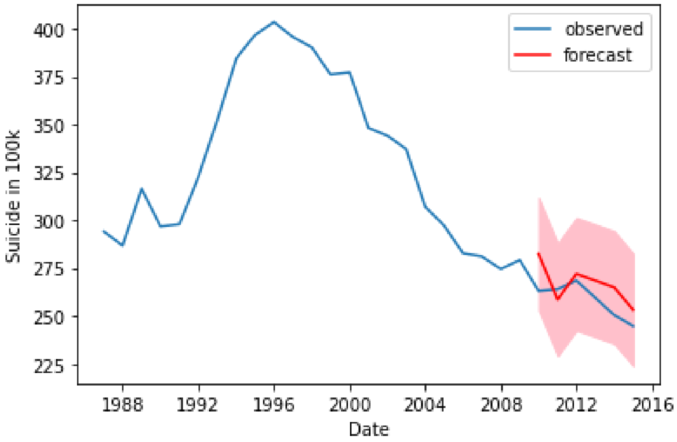
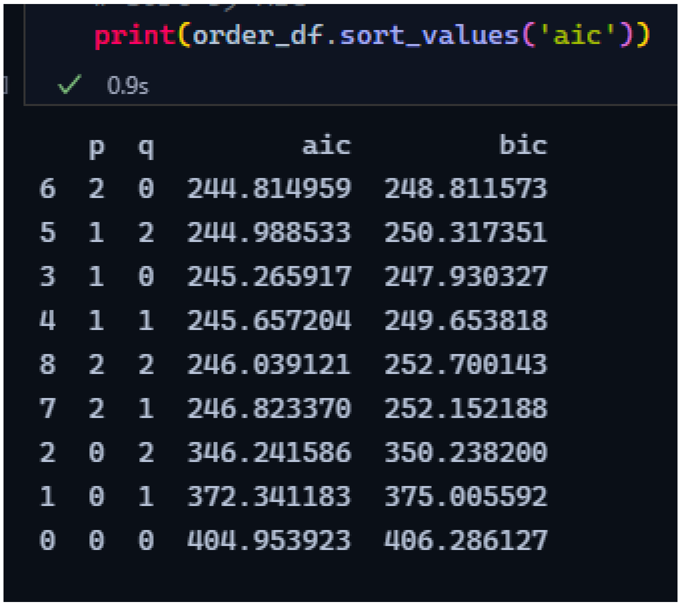


*Fig. 4.4**:** ARIMA Forecast and observed value graph on Russian data*

AIC and BIC graphs were made as shown in fig 4.11 for checking the order of the ARIMA model. In this research I am trying to work on predictions so, I will be looking at the AIC. A lower AIC score means a better predicting model. If the order is set too high, it could result in a high AIC value, this stops us from overfitting the training data. BIC is similar to AIC; lower BIC indicates a better model. BIC likes to choose a simple model with the lower order. AIC is better at predictive models, but BIC is choosing a good explanatory model.



```
print(order_df.sort_values('aic'))
```

✓ 0.9s

```
   p  q         aic         bic
6  2  0  244.814959  248.811573
5  1  2  244.988533  250.317351
3  1  0  245.265917  247.930327
4  1  1  245.657204  249.653818
8  2  2  246.039121  252.700143
7  2  1  246.823370  252.152188
2  0  2  346.241586  350.238200
1  0  1  372.341183  375.005592
0  0  0  404.953923  406.286127
```

Fig. 4.5: *AIC and BIC Score in ascending order on Russian data*

Here Fig 4.12, I am looking at a better predicting model so, I will be choosing AIC with the least score. This is an ARMA(2,0) Model

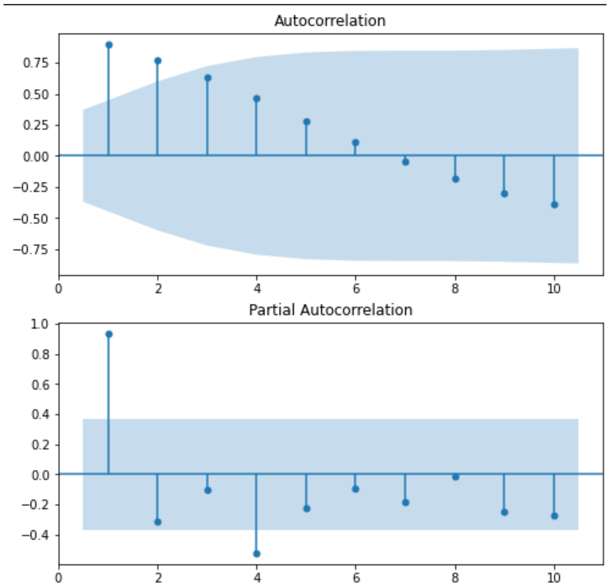
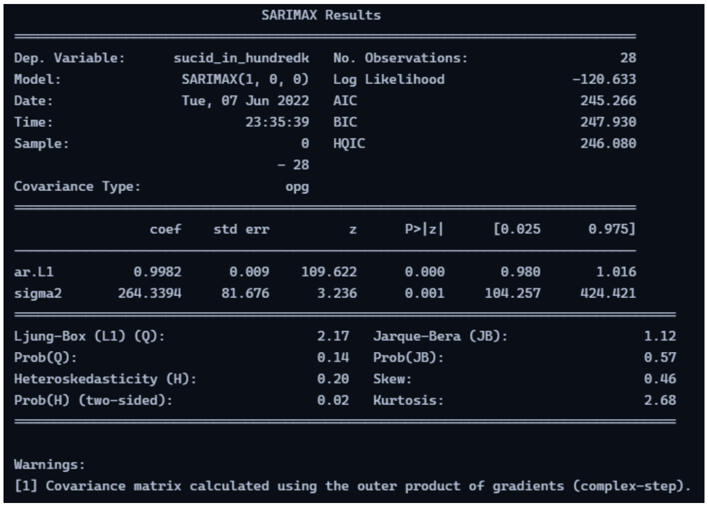


Fig. 4.6: *ACF and PACF to choose the model in ARIMA on Russian data*

From fig 4.13, In the above ACF and PACF, we can see ACF tails off and PACF cuts off since we have a MA(q) model. So this is an AR(1) Model.

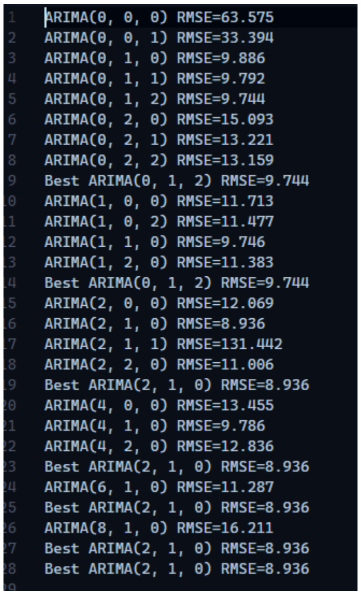


```
                               SARIMAX Results
==============================================================================
Dep. Variable:      sucid_in_hundredk   No. Observations:                   28
Model:               SARIMAX(1, 0, 0)   Log Likelihood                -120.633
Date:                Tue, 07 Jun 2022   AIC                            245.266
Time:                        23:35:39   BIC                            247.930
Sample:                             0   HQIC                           246.080
                                 - 28
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1          0.9982      0.009    109.622      0.000       0.980       1.016
sigma2       264.3394     81.676      3.236      0.001     104.257     424.421
===================================================================================
Ljung-Box (L1) (Q):                   2.17   Jarque-Bera (JB):                 1.12
Prob(Q):                              0.14   Prob(JB):                         0.57
Heteroskedasticity (H):               0.20   Skew:                             0.46
Prob(H) (two-sided):                  0.02   Kurtosis:                         2.68
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

__Fig. 4.7:__ Sarimax Model Results with order (1,0,0) on Russian data

4.4.1 grid search ARIMA parameters for time series

We can automate the process of training and evaluating the ARIMA Models on different combinations of model hyperparameters. In machine learning, this is called grid search.



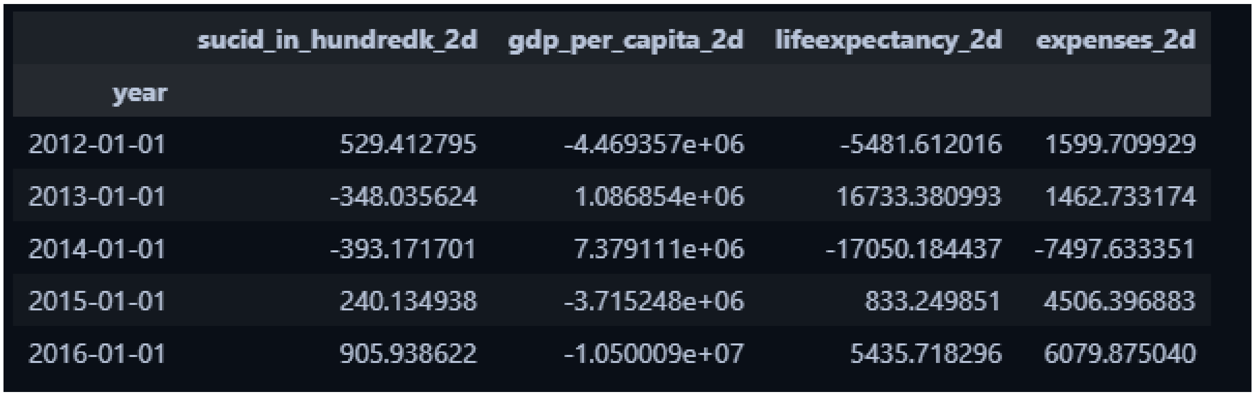
```
ARIMA(0, 0, 0) RMSE=63.575
ARIMA(0, 0, 1) RMSE=33.394
ARIMA(0, 1, 0) RMSE=9.886
ARIMA(0, 1, 1) RMSE=9.792
ARIMA(0, 1, 2) RMSE=9.744
ARIMA(0, 2, 0) RMSE=15.093
ARIMA(0, 2, 1) RMSE=13.221
ARIMA(0, 2, 2) RMSE=13.159
Best ARIMA(0, 1, 2) RMSE=9.744
ARIMA(1, 0, 0) RMSE=11.713
ARIMA(1, 0, 2) RMSE=11.477
ARIMA(1, 1, 0) RMSE=9.746
ARIMA(1, 2, 0) RMSE=11.383
Best ARIMA(0, 1, 2) RMSE=9.744
ARIMA(2, 0, 0) RMSE=12.069
ARIMA(2, 1, 0) RMSE=8.936
ARIMA(2, 1, 1) RMSE=131.442
ARIMA(2, 2, 0) RMSE=11.006
Best ARIMA(2, 1, 0) RMSE=8.936
ARIMA(4, 0, 0) RMSE=13.455
ARIMA(4, 1, 0) RMSE=9.786
ARIMA(4, 2, 0) RMSE=12.836
Best ARIMA(2, 1, 0) RMSE=8.936
ARIMA(6, 1, 0) RMSE=11.287
Best ARIMA(2, 1, 0) RMSE=8.936
ARIMA(8, 1, 0) RMSE=16.211
Best ARIMA(2, 1, 0) RMSE=8.936
Best ARIMA(2, 1, 0) RMSE=8.936
```

__Fig. 4.8:__ grid search result from ARIMA Model on Russian data

In fig 4.14, we can see I have implemented the grid search and evaluated the ARIMA Model. Also, I have evaluated a set of different parameters.

4.4.2 Prediction using Vector Auto Regression Models (VAR Model)

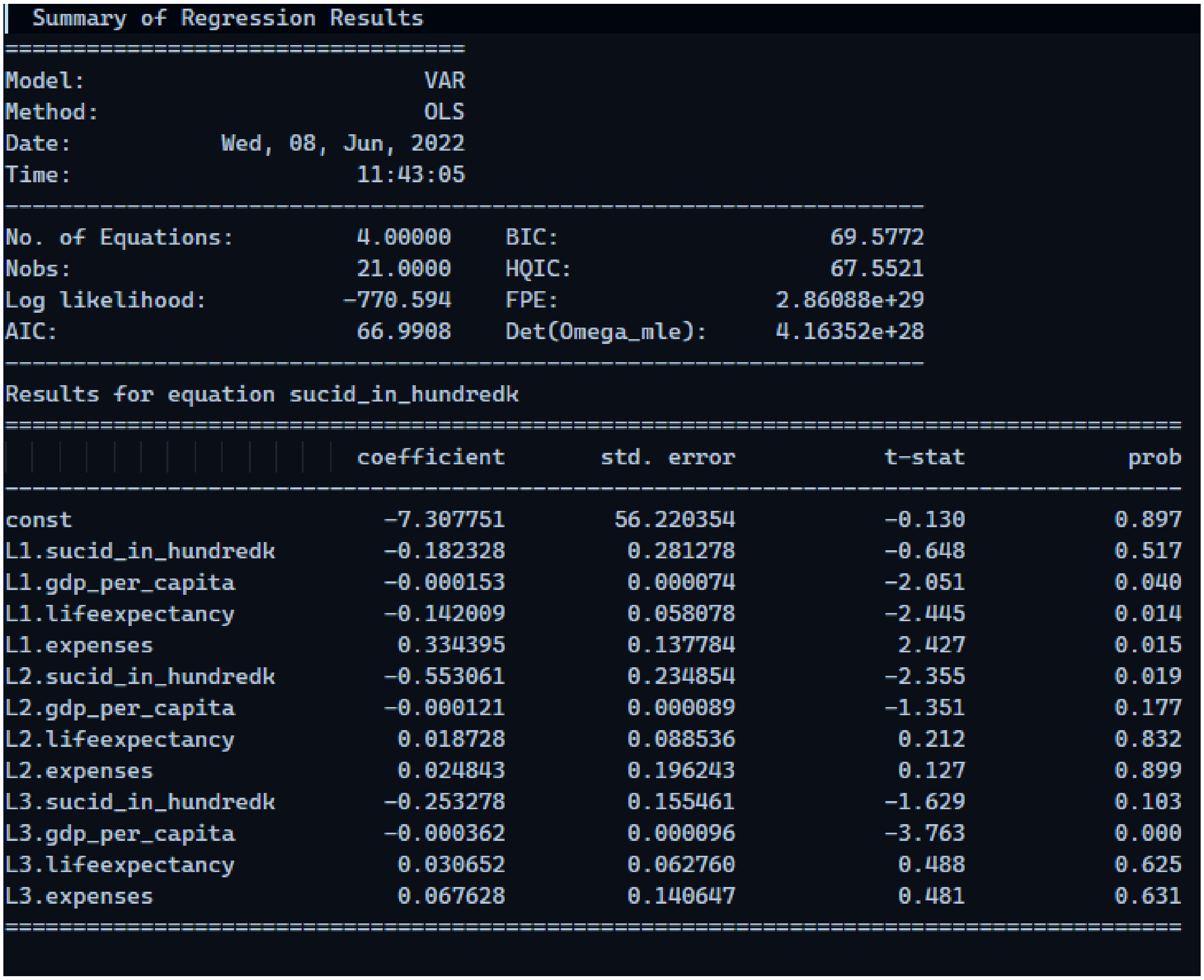
Another Model used for the time series data is the VAR model (Vector Auto Regression). The reason behind using this model is that it helps in forecasting models based on multiple variables in time series. Usually, we use single variable and sequential time for time series analysis. But here I was able to include multiple variables in the model as you can see in the figure. Vector Autoregressive Models are one of the best models we could use to choose for time series.



	sucid_in_hundredk_2d	gdp_per_capita_2d	lifeexpectancy_2d	expenses_2d
year				
2012-01-01	529.412795	-4.469357e+06	-5481.612016	1599.709929
2013-01-01	-348.035624	1.086854e+06	16733.380993	1462.733174
2014-01-01	-393.171701	7.379111e+06	-17050.184437	-7497.633351
2015-01-01	240.134938	-3.715248e+06	833.249851	4506.396883
2016-01-01	905.938622	-1.050009e+07	5435.718296	6079.875040

Fig. 4.9: *predicting future values in VAR Model on Russian data*

You can see in fig 4.15, that we have predicted the number of suicides for the year 2016 using VAR Model on the time series sequential data.



```
  Summary of Regression Results
==================================
Model:                         VAR
Method:                        OLS
Date:           Wed, 08, Jun, 2022
Time:                     11:43:05
--------------------------------------------------------------------
No. of Equations:         4.00000    BIC:                    69.5772
Nobs:                     21.0000    HQIC:                   67.5521
Log likelihood:          -770.594    FPE:                2.86088e+29
AIC:                      66.9908    Det(Omega_mle):     4.16352e+28
--------------------------------------------------------------------
Results for equation sucid_in_hundredk
=========================================================================================
                          coefficient       std. error           t-stat            prob
-----------------------------------------------------------------------------------------
const                       -7.307751        56.220354           -0.130           0.897
L1.sucid_in_hundredk        -0.182328         0.281278           -0.648           0.517
L1.gdp_per_capita           -0.000153         0.000074           -2.051           0.040
L1.lifeexpectancy           -0.142009         0.058078           -2.445           0.014
L1.expenses                  0.334395         0.137784            2.427           0.015
L2.sucid_in_hundredk        -0.553061         0.234854           -2.355           0.019
L2.gdp_per_capita           -0.000121         0.000089           -1.351           0.177
L2.lifeexpectancy            0.018728         0.088536            0.212           0.832
L2.expenses                  0.024843         0.196243            0.127           0.899
L3.sucid_in_hundredk        -0.253278         0.155461           -1.629           0.103
L3.gdp_per_capita           -0.000362         0.000096           -3.763           0.000
L3.lifeexpectancy            0.030652         0.062760            0.488           0.625
L3.expenses                  0.067628         0.140647            0.481           0.631
=========================================================================================
```

Fig. 4.10: *Regression result summary from VAR Models on Russian data*

The coefficient scores for multiple variables in the time series analysis are shown in Fig 4.16. Each variable's standard error is also included, and coefficient values reflect the correlation of variables utilized to create the model's statistical equation.

I was able to create predictions on multi-variate time series data using VAR Models, as I described previously. The actual value of and the anticipated value distribution in the VAR Model are shown in Figure 4.17.

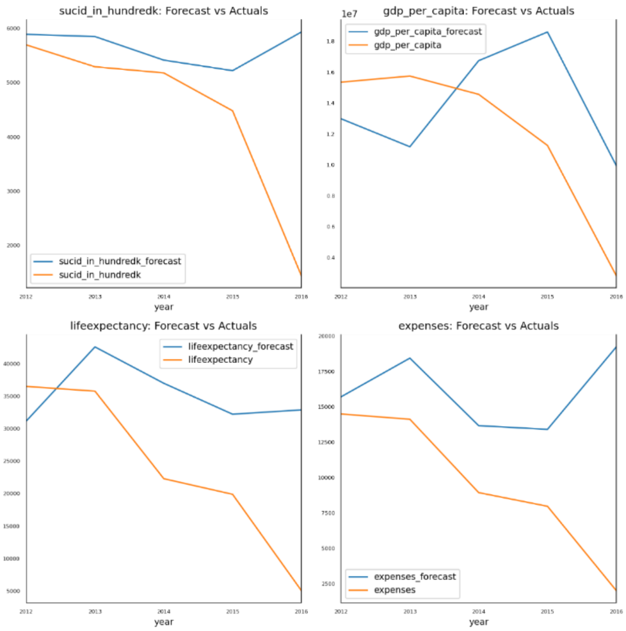


Fig. 4.11 *Plot of Forecast vs Actuals from VAR Models on Russian data*

From fig 4.17 you can see how the model forecasted different variables in the Russian suicide dataset concerning actual values.

4.4.3 Prediction using Auto Regression Models (AR Model)

The next model I have created is the Auto regression model, Train and Test were split into seventy and 30 per cent. Seventy per cent of the data was used for training the model and the rest thirty per cent was used for testing. I have got an 11.792 Root mean squared error. Also, I could save different models to the local and I was able to load the models later and update them accordingly.

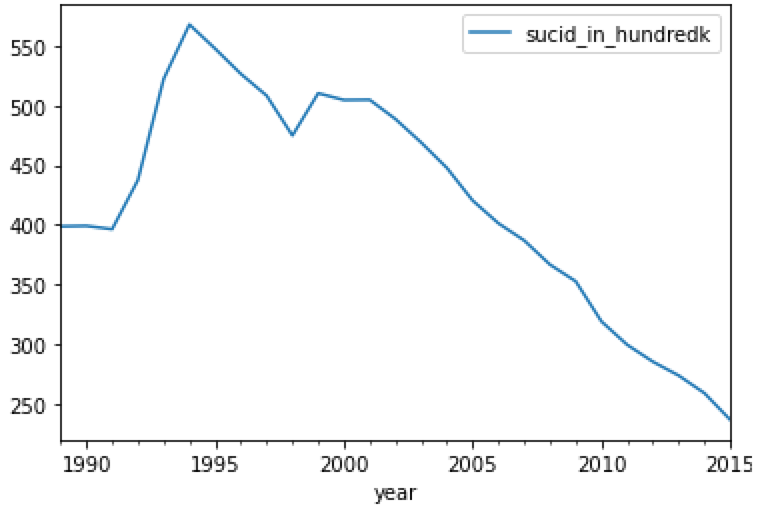


Fig. 4.12 Plot of suicide per hundredk on Russian data

As you can see in the diagram above, the suicides per hundred thousand are distributed throughout the year in 'Russian Federation' is shown.

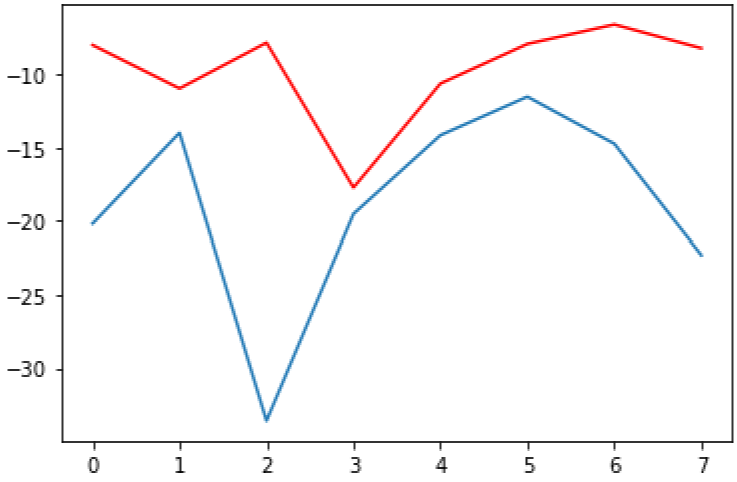
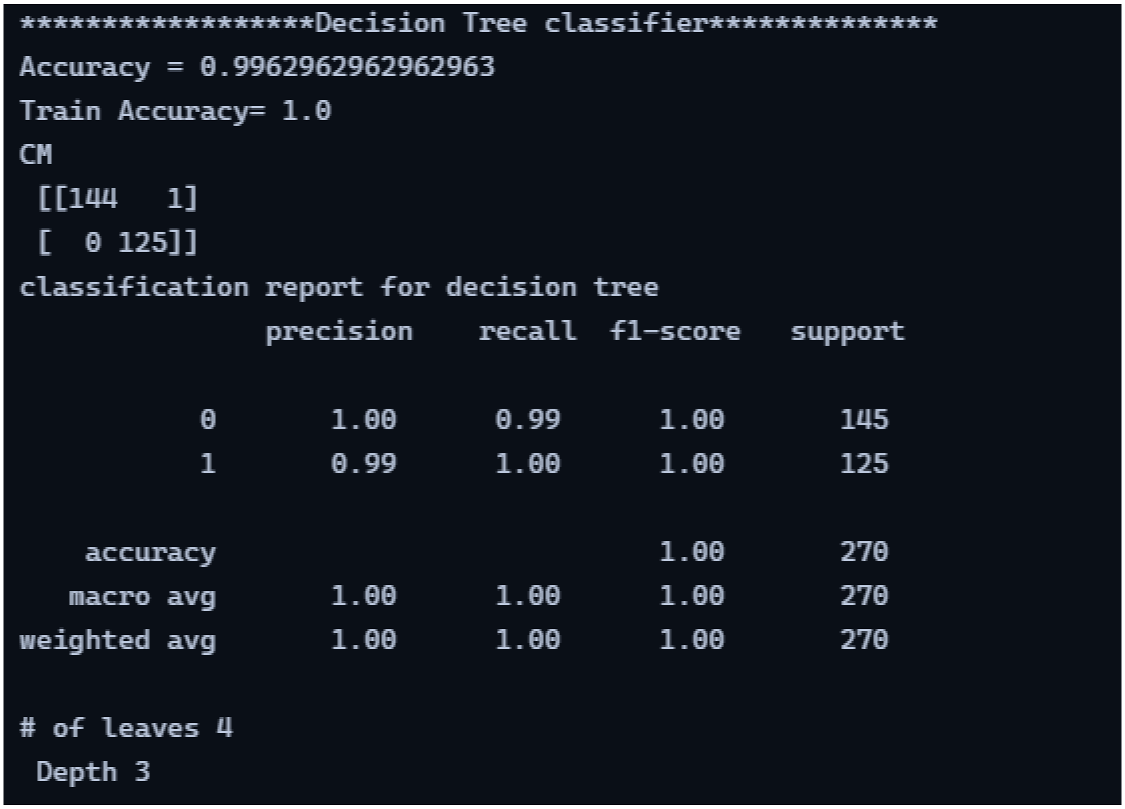


Fig. 4.13 Plot of Forecast vs Actuals from AR Models on Russian data

In this fig 4.18, the blue line is the test data and the red line is the predicted values. I have AR 1 Model.

4.4.4 Decision Tree Classifier

I have added a new column called risk where I split the data into two classes, class 1 stands for high risk and class 0 for low risk. Using the decision tree model, I have made a classification.



```
******************Decision Tree classifier**************
Accuracy = 0.9962962962962963
Train Accuracy= 1.0
CM
 [[144   1]
 [  0 125]]
classification report for decision tree
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       145
           1       0.99      1.00      1.00       125

    accuracy                           1.00       270
   macro avg       1.00      1.00      1.00       270
weighted avg       1.00      1.00      1.00       270

# of leaves 4
 Depth 3
```

Fig. 4.14 Plot of Forecast vs Actuals from AR Models on Russian data

Fig 4.20 shows the result from the Decision Tree classification algorithm. Lee and Oh (1996) have done studies on Neural networks using a Decision Tree classifier to distinguish between complex features. Here this goals are to separate the risk group and non-risk groups based on the feature called risk. I got 99.62 training accuracy and a hundred per cent testing accuracy.

4.5 Data Storage

Thirdly, we need a database server for data to be stored on the server. I will be using PSQL or ThisSQL servers for data storage and management. I want the data in this DB to be updated from time to time and this model must be updated based on the new data injected in each time. The reason for choosing these DB's is the flexibility of usage and its syntax matching with Structured Query Language (SQL) minute differences. Initially, data was stored in the CSV format in different files. Later I uploaded them into Mochahost Psql server.

4.6 Web Server and Hosting

We know there are thousands of hosting companies providing hosting services, I have chosen Mochahost as one of the best service providers for small business websites. This goal is to make a highly dynamic web application on the server. I have purchased a VPS service which allows running PIP Packages on their server making the server IDE more suitable for the Dash App. Mochahost cPanel will be connected to the Github repository where this application is updated from time to time. Using git technology for the hosting makes the process more sophisticated and professional in terms of version control.

Chapter 5

Evaluation and Testing

Evaluation of the model is as important as making the model. I have created 3 models in ARMA, Auto regression and Vector Auto regression. Using mean squared error and R-squared error I took the error rates of different models. Accuracy is also calculated to understand how efficient and precise this model is. Despite the fact that the model below was constructed using a Russian dataset. The live dashboard and modeling for the final product will be based on the whole country's master dataset.

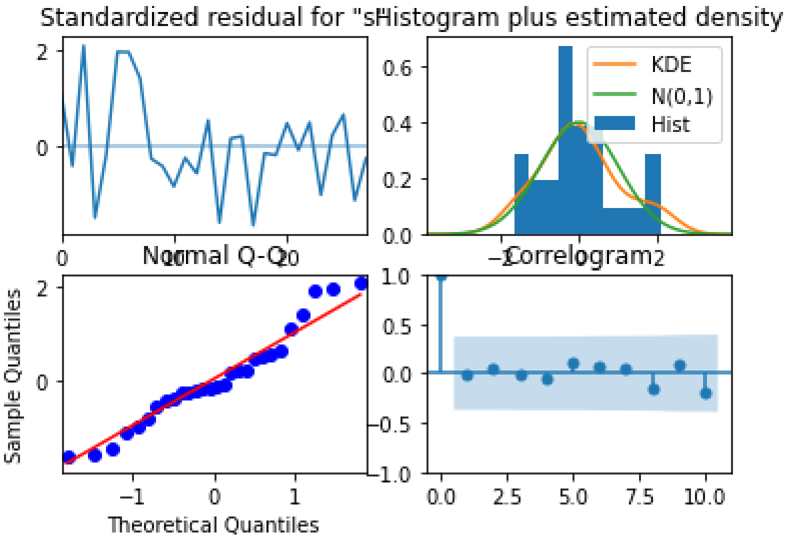
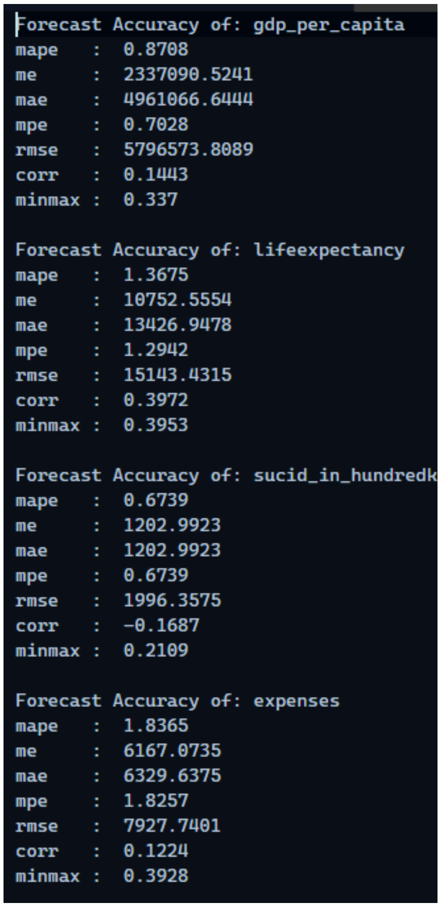


Fig. 5.1 *ARIMA Diagnostic plots on Russian data*

After finalizing this model, I would be able to add more evaluation techniques. Above are the four diagnostic plots I have created after running ARIMA Model. Looking at the plots I can say there is no pattern in the standardized model. Looking at the Histogram there is no Gaussian distribution (green and red lines should be almost the same for gaussian). The QQ Plot seems to be not normally distributed, if it is normally distributed all the blue dots will be aligned over the line (except for some values in either end).



```
Forecast Accuracy of: gdp_per_capita
mape   :  0.8708
me     :  2337090.5241
mae    :  4961066.6444
mpe    :  0.7028
rmse   :  5796573.8089
corr   :  0.1443
minmax :  0.337

Forecast Accuracy of: lifeexpectancy
mape   :  1.3675
me     :  10752.5554
mae    :  13426.9478
mpe    :  1.2942
rmse   :  15143.4315
corr   :  0.3972
minmax :  0.3953

Forecast Accuracy of: sucid_in_hundredk
mape   :  0.6739
me     :  1202.9923
mae    :  1202.9923
mpe    :  0.6739
rmse   :  1996.3575
corr   :  -0.1687
minmax :  0.2109

Forecast Accuracy of: expenses
mape   :  1.8365
me     :  6167.0735
mae    :  6329.6375
mpe    :  1.8257
rmse   :  7927.7401
corr   :  0.1224
minmax :  0.3928
```

Fig. 5.2 *Plot of Forecast vs Actuals from VAR Models on Russian data*

As you can see in fig 4.22, we have calculated the accuracy of each variable. This helps us understand how well our model is performing.

In the dash app, user can pick any model from the dropdown list and see the RMSE score for the respective model. Also, the score may vary based on the country name as well. Seeing the RMSE scores, it's easy to say that three of the models including FB Prophet, Custom Auto Regression and SARIMA performed very well. Out of the three models SARIMA has made best predictions compared to the other two.

Model Name	RMSE Score
SARIMAX	12.88
Custom AR	16.89
FB Prophet	24.75

Table 5.1: RMSE Scores on predicting suicides with different models for Ireland

5.1 Applications and software

The most popular IDE for coding is Microsoft Visual Studio. I've been using Github for version control. In addition to Jupyter Notebook, Spyder, and Atom, I've used various coding tools such as XAMPP, HeidSql, Postman, Putty. All the testing is done in the Anaconda environment on the local machine. For the whole study, Python version 3.8.8 was employed. The PIP package is needed to set up the IDE. Minor CSV file checks are also performed using Microsoft Excel. The built-in git version control facilities in Visual Studio are occasionally used to manage the repository's branches. Microsoft Visual Studio is the main IDE used for coding. For version control I have used Github. I also have used other tools like Jupyter Notebook, Spyder, Atom for coding purposes. All the testing are done with the local Anaconda environment. Python version 3.8.8 is used for the whole research. PIP package is used for configuring the IDE. Microsoft excel is also used for minor csv file inspections. Visual Studio’s inbuilt git version control features are used time to time for managing the branches in the repository. MS Word and notepad are used for reporting and notes. PowerPoint is a presentation software that allows you to create slides. PDF files are managed with Adobe Acrobat DC. !pip and git commands are run via the built-in terminal in Visual Studio, Anaconda Prompt, and Windows Terminal. The entire project is run on the Windows operating system. The browsers utilized in this experiment are Google Chrome and Mozilla Firefox.

5.2 Ethical Considerations

To begin, consider some recent unethical events. When a researcher's action has a negative impact on participants or society, it is considered harm. There are a number of reasons why researchers' actions cause so much harm for society or people. Likewise, one of the most obvious examples of such accidents was Human Radiation Experiments (2017) was one of the biggest examples of such incidents. In 1994 US President Clinton created an advisory team to research human radiation that has been conducted over the years. In this study, doctors injected Plutonium into the body of many patients and many of them did not consent to be part of this study. Also, there was a company called Quaker

Oats (2020) which is also part of this study included radioactive components in oatmeal and were unknowingly fed to the children.

In this study, no such experiment is done on humans in the process of data collection or analysis. An aggregated suicide dataset only provides information about the country's general population and related detail as features is used throughout the research. No prior experiment is conducted to gather data for this research. No harm is made to any subject in this regard. There are several benefits related to the data. Data provides an overview of how many suicides are happening from time to time. Talking about the societal impact of this research is enormous. For example, Study of Benefits of Electric Cars (2016) has created a significant impact on how this research has benefited society to help understand the carbon footprint reduction and cost-saving. In suicide analysis, I am trying to make use of data to leverage suicide attempts by helping the government to take measures or policies from the outcome of this study it’s going to help create plans to tackle such acts in coming years.

This study of suicide analysis was based on a dataset that is open source and available for download on Kaggle. In terms of data storage and security, I wouldn't call it a particularly sensitive dataset since, for starters, this information is not private on the internet, and the creator has left public access open. Second, this suicide dataset does not contain any personally identifiable information; rather, it is a summary dataset that provides broad statistics on the country's mortality rates. Also, information on who is more prone to commit suicide, such as age group, internet users, human development index, and so on.

5.2.1 SWOT Analysis

They may eventually take us to court is one of the real-life examples where in 1942 prisoners were asked to undergo dangerous experiments to understand the survival chance of soldiers sometimes even leading to deaths. Understanding personal, social, and business impacts of data practice.

In addition, even sharing information of individual sharing with any other colleagues or third party would be through proper procedure and getting signs on consent forms.

Strength: - In this study, I am trying to see suicide rates in different countries from time to time. This research strength is its dynamic nature. Similar to the weather forecast of google or Microsoft, this model will be run from time to time based on the latest data. This research aims at tackling suicide tendencies in every country's population. This research is going to predict how many people are going to commit suicide in the next 5 years in different countries or continents. When working with a socially responsible research project, it is going to stand out in the world of the internet. Similar to the websites showcasing covid trends live, this website is also going to show the same impact of suicide numbers and create respective visualizations for any general audience to easily understand what the trend in data would be.

Weakness: - For forecasting, the data is aggregated, and no unique information about individuals is provided. As a result, I believe the data must have more precise traits that may be used to create reliable suicide predictions. However, if additional characteristics could have been added to the dataset, the model would have been more accurate. Things like topic diagnostic information, population happiness index, education index, and each country's happiness index. As a result, this forecast is more of a broad grasp of data trends.

Opportunity: - It's difficult to put into words how much can be learned through suicide data analysis. The government is working to figure out what causes suicides and how to reduce the number of suicides each year. We can build creative strategies to lessen the effect of suicides by studying and interpreting current statistics. Machine Learning models might be used to develop smart apps that help mobile users based on their activity data. Suicide analysis ushers in a new era of artificial intelligence in which we can track who is on the verge of dying.

Now let's look at this data and its opportunities. Have you ever thought of having a suicide prediction model for each country? The wide range of opportunities using AI and the Time Series model on big data is possible using current technologies. Internet of things, cloud computing and Machine Learning are the best examples of state-of-the-art technologies.

The suicide prediction model and live dashboard visualization is a great analysis model which any growing business can take inspiration from. Just imagine a burger selling vendor creating a live predicting model of a specific kind of burger that is sold at a particular season of a year? or maybe checking bestselling milkshakes each month? Wouldn’t these analyses make them grow? or even predict how many products are going to be sold in the coming months so they can prepare their store for the coming period to avoid lack of materials. Thus, this model is ultimately showing what kind of predictions or analysis our business and health industry need today to go smarter and do smarter businesses.

Threat: - Data may be utilized in a variety of ways. Some individuals utilized it for good causes, while others exploited it in a different way. It's possible that the suicide data will be abused in some way. However, from this perspective, they are less likely to occur if we do not provide detailed information about people. In this scenario, I'd argue that if new features are added to the model in the future, I'll have to change the model statically and then make it dynamic using cron tasks. Furthermore, additional storage space may be soon necessary when it comes to keeping individual information, and this model may function badly due to server needs. Even if we have alternative possibilities for purchasing cloud storage space, it will still be more expensive; still, I will have to find ways to enhance the needs. When it comes to the examination of prior years' suicides, the differences in counts over different political administration eras might have a political influence.

5.3 Data Security

Data Security has become an important concern in this era. Even though the suicide dataset is publicly available on the internet, I have followed the best practices in data security to ensure there is no data leakage. I have used encrypted windows drive to store the data. Whole project codes are updated from time to time to GitHub private repository. Any information related to this study has been considered for data security and ethical practices before using them. No personal information is used in this study. For web applications, files are kept in a private repository and used that repo to pull changes to the live mocha host server.

5.4 Proposed Future Analysis

The preliminary analysis of this research reveals that additional specific data is necessary for the project. In order to produce more accurate and meaningful forecasts in the future, daily suicide data with more precise personal information will need to be obtained. Future development will involve expanding the backend options to include a form with more fields particular to the person who died so that individual data may be collected. For instance, the person's blood group, health information, alcohol use, and so on. More models, such as SVM, will be used in future study to assist us understand the best approaches for forecasting and achieving optimal results.

Chapter 6

Results

Time series forecasting is a challenging process with no simple solution. The optimal statistical model is never quite evident because there are so many models that each claim to perform better than the others. Because of this, ARMA-based models (Brendan Artley, 2022) are frequently a suitable place to start. They are suitable as a baseline model in any time series problem and can produce respectable results on most time-series problems. Models such as ARIMA, VAR, and AR were utilized to investigate and forecast the effect of suicide in different nations in this work. All the models were built using the 'Russian Federation' suicide dataset as a starting point. When compared against other algorithms, the ARIMA model outperformed the others. One of the models, called SVM, has been found as a good fit for working on the suicide dataset, which must be done alongside other models that have already been completed. In comparison to other models, ARIMA and Decision tree classifiers provided me with greater accuracy. From this point forward, I'll be working on the primary dashboard, which is an online tool that makes real-time forecasts based on data input.

The other two models used in forecasting was FB Prophet Model and SARIMA both of which gave very good results in this study. As per the progress seen by 2025 the suicide rate will become zero in Hungary. FB prophet predictor also forecasted negative values in the graph shows that the model must be empirically redesigned according to real life suicide figures.

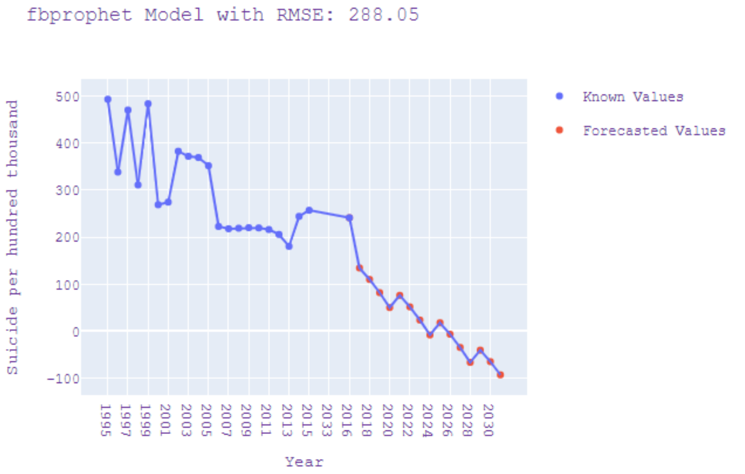


Fig 6.1 Prediction of Hungary for next ten years using FB Prophet

SARIMAX model on the other hand given prediction with 49.27 RMSE score. SARIMAX predicted with not much change in the suicides in the coming years. The model prediction is also showed strange steady prediction which also quite different from the fact that suicide wouldn’t be same all the years. In case of Ukraine, the RMSE score for SARIMAX Model was 110.15, but FB Prophet and Custom AR has given 145.65 and 229.53 respectively. In Ireland, Data SARIMAX Gave 12.88 RMSE score on the other hand, FB Prophet and Custom AR models gave 24.75 and 16.89 respectively.

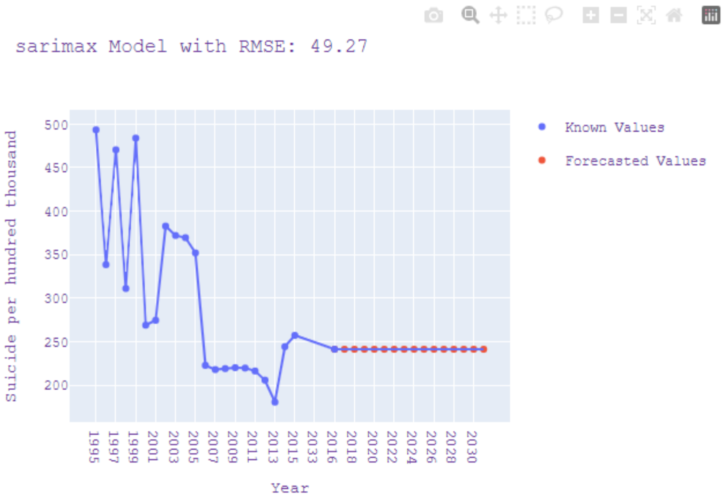


Fig 6.2 Prediction of Hungary for next ten years using SARIMAX

Data integrity check is a very effective technique to visualize the outliers in the data. Periodic outlier detection was the technique used to identify the data integrity. Countries like Ireland showed cleaner data compared to most other countries. From the above discussion it’s clear that, data of Ukraine and Russia must have more outliers which was also visible in this graph. data_modelling.py is run to make the output.csv file which would be later used to visualize the outliers. An unsupervised ML algorithm called DB Scan is used in making the outlier detection. This is to show the app visitors that, how much outliers are found in the updated data every week since new records are added from backend time to time.

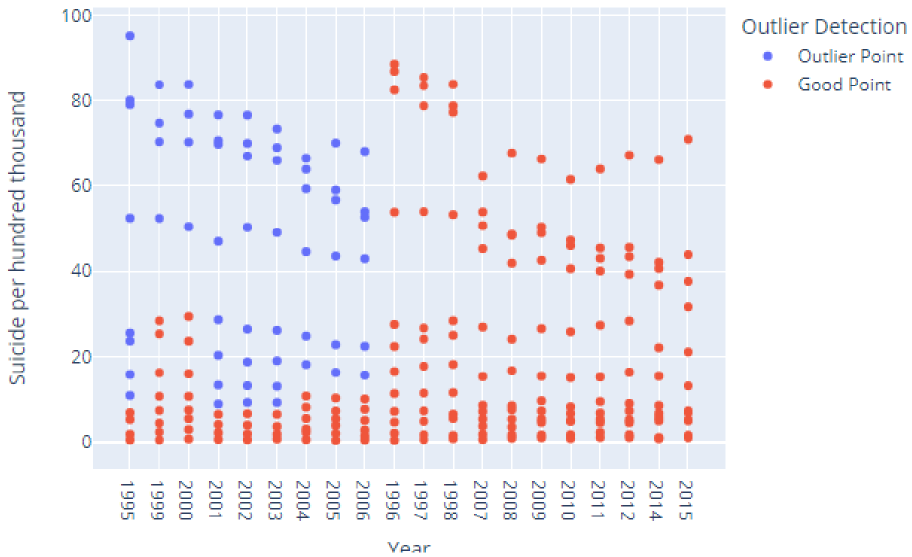


Fig 6.3 Outliers detected in Ukraine data

As per the fig 6.2, the suicide rate is going to decrease until 2030. The results show that most of the suicides happened between the age of 35-54 years of age. And second most affected age group is between 55-74 years old. Least affected age group are between 15-24 years old. In all the age groups, Male population is the most affected age group.

Most European countries showed high GDP per capita this shows that too in rich countries and too poor countries suicide rate is too high. Few examples of most affected countries are Hungary Russian Federation and Ukraine. Countries like Armenia, South Africa.

Moreover, out of all these countries Hungary has the greatest number of suicides ever in 1991 with 575 suicides per hundred thousand. As per the study (Havasi et al, 2005) showed that five hundred and one (69.6%) of the 719 suicide deaths overall were caused by men, while 218 (30.4%) were caused by women. Men's largest age categories were 41 to 50, while women's largest age groups were 41 to 50 and 71 to 80. Hanging was the most popular method of execution (46%). The findings showed that 38.8% of the 474 victims whose blood and/or urine alcohol content measurements were made previously drank alcohol. A case-control study (Almasi et al, 2009) showed that In Hungary, the suicide rate is declining. Numerous risk factors connected to individual-level demographic and clinical traits as well as potential recent societal change were found in their study. Self-harm and psychiatric disease management that is improved could lead to more declines in suicide rates.

Meanwhile in Ireland, there has been an upward trend from 1985 to 1998 where the suicide rate has changed from 102.16 suicides per hundred thousand to 162.17 suicides per hundred thousand. From the year 2000 Ireland suicide rate has fallen. According to the results of the custom auto regression model shown in Fig. 2.1, the number of suicides per 100,000 people is expected to rise during the next fifteen years. Ireland is estimated to have 136.81 suicides per 100,000 people by 2029, allowing us to start taking preventative measures before a worse situation arises.

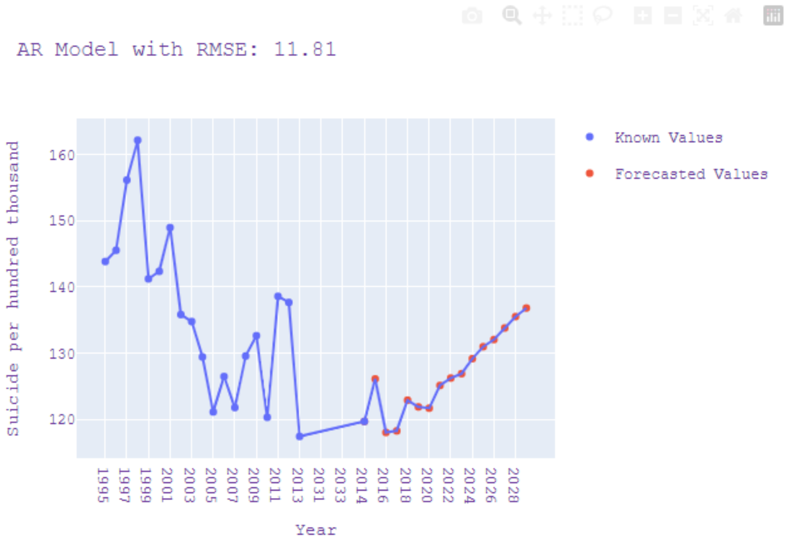


Fig 6.4 Predicting Ireland’s suicide in hundredk using Custom AR

On the other hand, FB prophet model showed prediction with Root Mean Square error 24.75. After 2014 it seems to be having a quick rise in the number of suicides per hundred thousand. In 2015 the suicide rate is given 146.61 and from there it shows a slight upward trend. As the last prediction point year 2029 shows 148.63 suicide rate which is far away from what we got from the custom auto regression model.

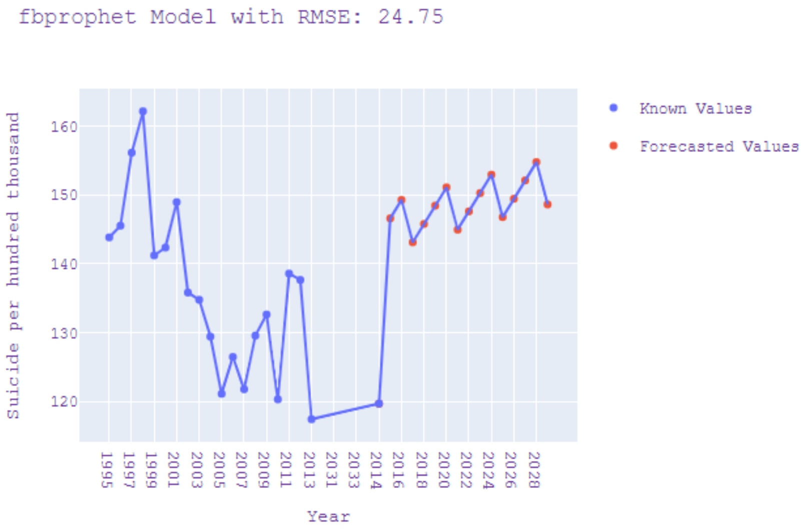


Fig 6.5 Predicting Ireland's suicide in hundredk using FB Prophet

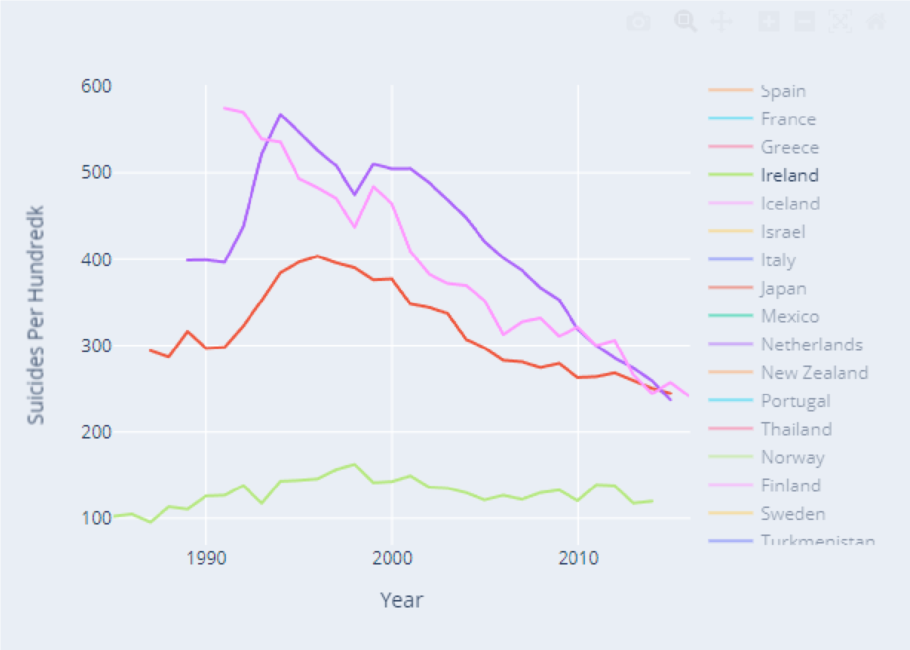


Fig 6.6 Suicide Per Hundredk in Ireland, Russia, Ukraine and Hungary

Comparing the situation in Ireland, Ukraine, and Russia we can see Russia has had the greatest number of suicides per hundred thousand since 1994. The reason for taking these countries is because as stated in the above paragraph these are the countries with the highest suicide rate. As given in fig 6.4, the suicide rates have a huge change over the decade in these countries. In Ukraine from 1991 to 1996 suicide rate has a huge increase. From 1996 to 2015 it has dramatically decreased from 403.42 to 244.72 suicides per hundred thousand. In Ukraine, as per research done by Nordstrom (2007) shows that Among heavier drinkers, a strong correlation exists between alcohol use and suicide or attempted suicide. Psychiatric comorbidity (WHO, 2005) in this population raises the chances of suicide behavior. Between 1985 and 1988, the USSR waged a vigorous anti-alcohol campaign, which saw significant drops in mortality and alcohol consumption. "…a lot of penniless people were wandering the streets, people who had been well-educated, respectable citizens but who had lost their jobs and taken to drink when they could find no place for themselves in the new reality"(Anna Politkovskaya, 2004). As per the report from James Watkins (2017) Alcohol usage was also a major problem in the male population. According to WHO men's suicide rates are more than six times higher than women's rates in Russia, Ukraine, Belarus, Latvia, Poland, and Kazakhstan.

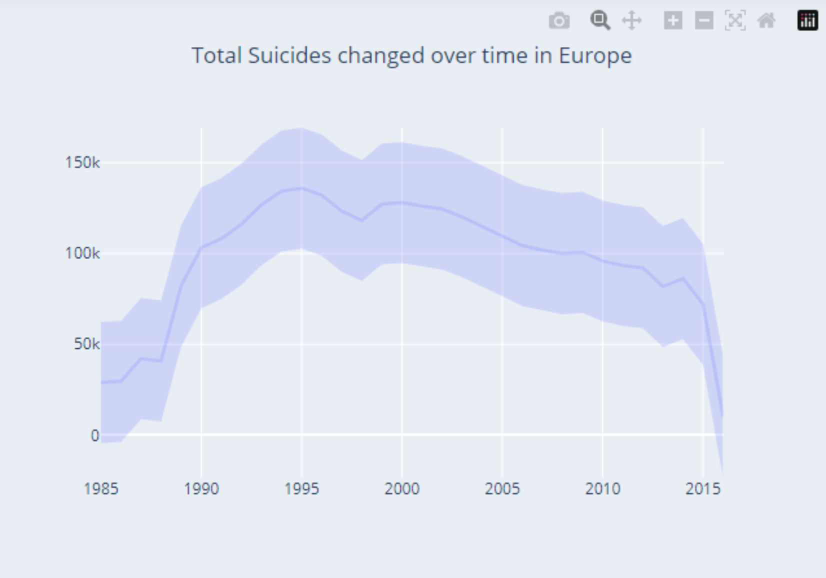


Fig 6.7 Total suicides changed over time in Europe

Overall suicides in Europe have changed positively for society. It is also visible from the above examples is that countries' economic situations drinking, and lifestyle also highly affected in increasing suicide numbers. It's interesting to see in almost all the countries how much of deaths happened in the male population. It is also understood that men are more vulnerable to suicide because they also look after the country like working in military or working for family, thus they are taking more effort run a comfortable life which make them depressed due to many situations in each country. For example, In Russia most common issue addressed in terms of suicide was alcohol consumption. Also, men working in the military has committed suicides (Joel Gunter, 2022). The final dashboard made is more of storytelling way and simple for any audience to understand the finding's discussed throughout this paper.

Chapter 7
Conclusion

A suicide analysis was carried out to know the reasons behind thousands of suicides happening around the world each day. The CRISP-DM approach served as the foundation for the methodology used for this project. Detailed research was carried out before starting the initial data load and analysis. Having many research questions and project goals, a proper dataset with records of all the countries around the world was required for this project to begin. After getting the first dataset from Kaggle analysis has begun with the help of Jupyter Notebook IDE. Initially loaded dataset was prepared by cleaning and removing outliers. The Final cleaned data set was then used for getting insight from the data using Exploratory Data Analysis (EDA) as the second step. Following that, the seaborn and matplotlib libraries in python were used to explore the datasets. As per the objectives drawn in the initial part of the research paper, variables like suicide per hundred thousand, age, sex, GDP per capita etc. were thoroughly considered in EDA. Russia, Ukraine, and Hungary are the countries with the highest suicide rate. People with age thirty-five to fifty-four years old are the most affected age group in terms of suicide per hundred thousand. Throughout the study suicide per hundred thousand was given more importance than suicides because countries with high populations could show high suicide numbers, then the scale of suicide figures will mismatch and does not make sense for this research. So, suicide per hundred thousand has been taken as the target variable for this research.

Several Models including SARIMAX, Custom Auto Regression and FB Prophet was used in forecasting. All models predicted suicide per hundred thousand variables over the next

10 years. RMSE Score has been checked on each model separate. Vector Auto Regression Model was only used in the initial analysis. VAR Model has been removed from further analysis because of the poor results found during the evaluation.

The learning from the project cannot be quantified because several different methods and methodologies have been used in this iteration. Using an advanced and most modern framework called Laravel gave an extra feature to this dash app. Python framework called python dash using plotly has been beautifully designed to showcase the insight from data and manage to host it online. Usage of data architecture skills in this project was inevitable. Data has been loaded to MySQL maria DB server using some commands and though the project data used was loaded from the DKIT database. Using Cascade Style Sheet (CSS) framework called bootstrap being one of the top-rated User interfaces (UI) in 2022 has also been used to design the dashboard as well as the feedback section.

Another important feature was using migration files in the Laravel eloquent package to make the Database filled with tables required for the data upload. Data migration and seeding of admin panel credentials are also done by the Laravel package without needing to manually enter them each time inside the database. Laravel uses a file called '.env' for setting environmental variables, it also follows the best practices in PHP so that any other person who will be working on the app in the future would be able to handle the code very easily. In any development project git version control is an inevitable technology. During the creation and development of this python dashboard, GitHub has been used all the changes have been documented as commits in the repository. An important aspect of this project was its ability to make a prediction based on the new data added from backend time to time, that is the reason for bringing cron jobs for this project to make the python files run once every week.

However, due to the insufficient data, it was not possible to suggest what is the core issue leading people to commit suicide. Though this study has many limitations at present adding more complex variables to the existing database and making use of some more supervised machine learning algorithms, the app could be improved in future.

For future work, it will be interesting to collect personal information such as record track of alcohol consumption, medical history including physical and mental health records of everyone with their consent so that study could bring out more precise reasons that lead people to think of committing suicide. Also, the server capacity must be upgraded with a better VPS package which could potentially carry such a vast amount of data that going to be added through the Admin Panel Backend. Another improvement on the app will be adding more features to the admin panel including the number of cases yearly, country-wise displayed on the admin panel home page along with a couple of graphs that could visually communicate the trend in suicide along with the forecast or prediction. Cron jobs in the future must run at least once every week. Also, a notification is to be shown on the dashboard page when the model predicts suicides in the next 10 years is showing a high level so that the visitors get an alert of future events in the world.

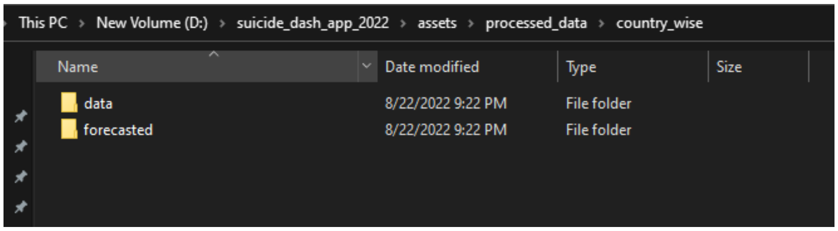
7.1 Limitations on Suicide Analysis

Each suicide that occurs in the world has a different set of causes since there are thousands, if not millions, of reasons why people commit suicide. A dataset that has already been aggregated won't produce empirical findings. Personal data must also be gathered and examined to draw more specific conclusions about the causes of those fatalities.

1. Country Names Missing: The analysis would have been more insightful if there had been data available from every country in the world as there are many nations missing from the dataset.
2. Server Requirements: Handling such large dataset requires more server capacity run the model time to time. Current server is only capable of running models with a smaller number of records, so the project is made in such a way that I can be run only once in every week.

8.0 Deployment

Python Dashboard has been deployed to Mochahost server initially. Getting server ready with required python package is very important step in the deployment stage. Several blogs regarding the python dash app (2020) have been referenced in order to figure out what is the procedure for hosting the python dash app.



This PC › New Volume (D:) › suicide_dash_app_2022 › assets › processed_data › country_wise

Name	Date modified	Type	Size
data	8/22/2022 9:22 PM	File folder	
forecasted	8/22/2022 9:22 PM	File folder	

Fig 8.1 Data Storage location in Local Machine

The first step is to run the data_maker.py file which takes the data from the assets/data folder. Raw data is stored inside this folder where the data_maker initially prepare the data and merges the dataset and exports to the output.csv file making it ready for further processing.

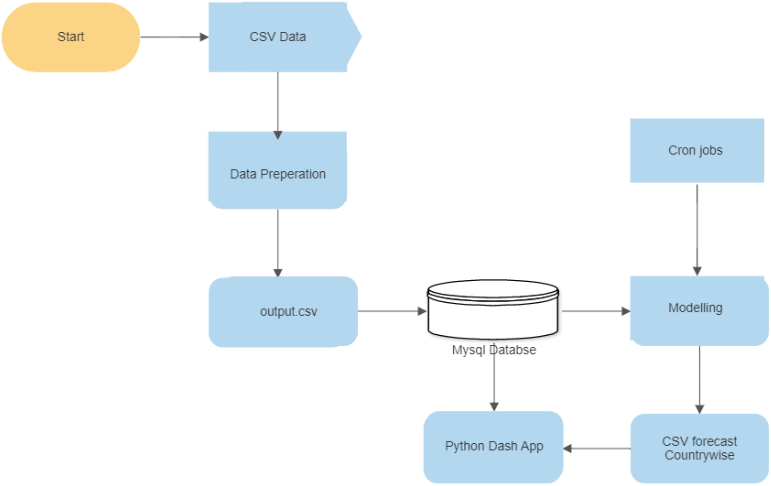
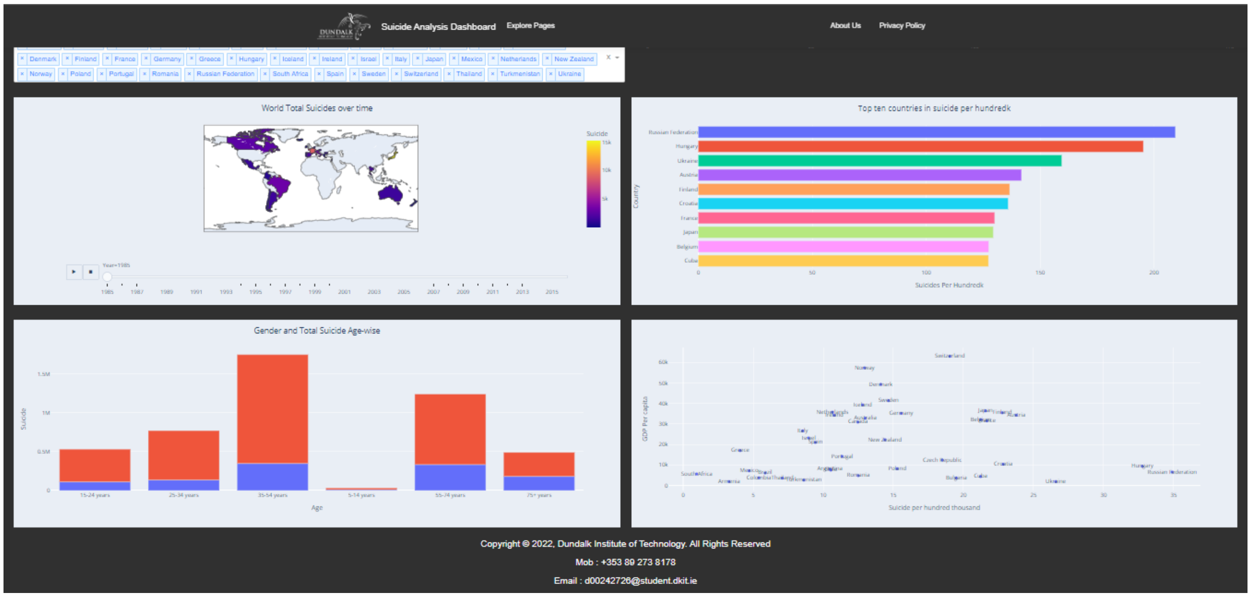


Fig 8.2 Structure of the dashboard

Second file called data_modelling run the models using output from first file mentioned above to create models and make forecasted and normal data based on each country name from the data available. These country wise files then are loaded into the dashboard pages using normal python dashboard programming and forecasting is made accordingly.

On the other hand, all the other graphs will be made with real time data loaded from the MySQL database to make live visualisations in the frontend. Initially python dash app is pushed into the GitHub and Heroku repositories to maintain the version control technology. Then the repo has been cloned into the Mochahost server in order to run the app. The process is shown in the fig 8.2.s



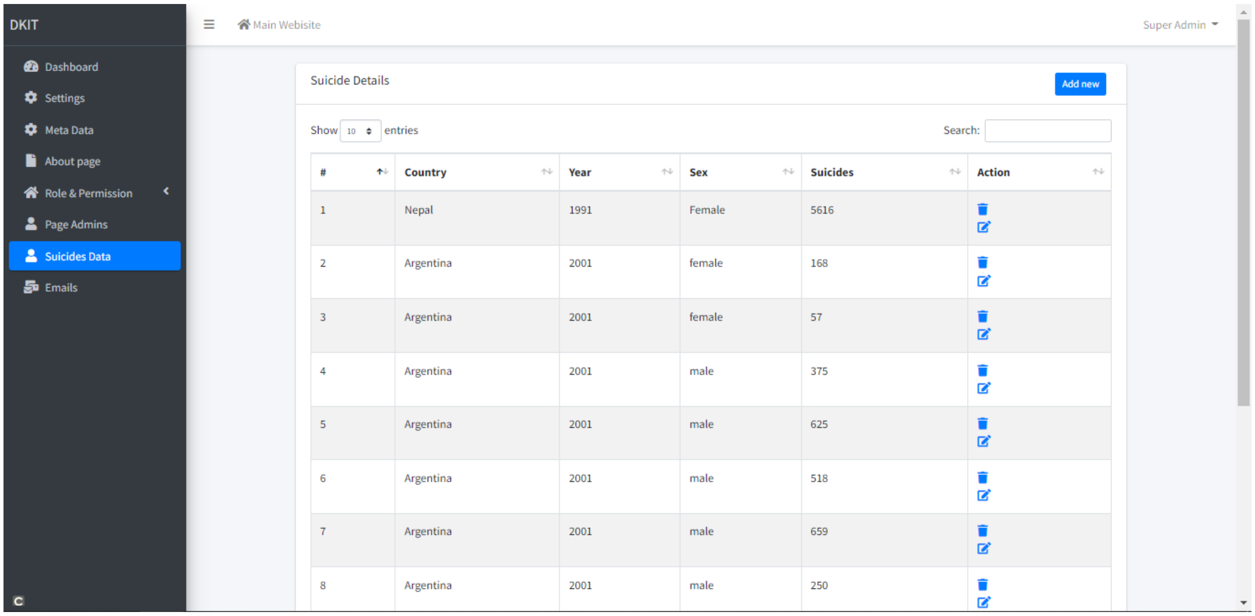
Copyright © 2022, Dundalk Institute of Technology. All Rights Reserved

Mob : +353 89 273 8178

Email : d00242726@student.dkit.ie

Fig 8.3 Dashboard after deployment

Admin panel backend section done in PHP Laravel the second hosted app. It shares the same MySQL database to keep the records UpToDate. So, as we move on, new suicide records around the world will be recorded in the admin panel suicide records section. One of the books helped in the development of the admin panel is called 'Mastering Laravel' (Pecoraro et al., 2015). Proper research concerning the security features in the Laravel framework (Aborujilah et al., 2022) compared with other popular ones has been referenced before choosing Laravel for safely data handling from the backend.



Suicide Details

Show 10 entries

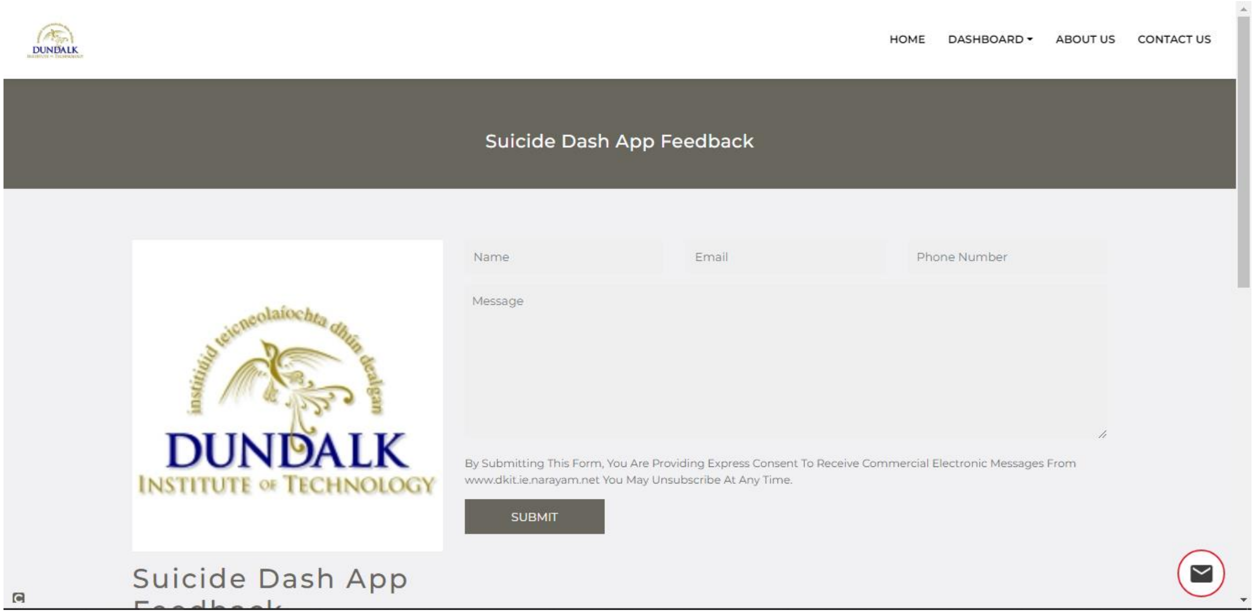
Search:

#	Country	Year	Sex	Suicides	Action
1	Nepal	1991	Female	5616	🗑 ✎
2	Argentina	2001	female	168	🗑 ✎
3	Argentina	2001	female	57	🗑 ✎
4	Argentina	2001	male	375	🗑 ✎
5	Argentina	2001	male	625	🗑 ✎
6	Argentina	2001	male	518	🗑 ✎
7	Argentina	2001	male	659	🗑 ✎
8	Argentina	2001	male	250	🗑 ✎

Fig 8.4 Admin Panel section for managing the dash app data

SSH management on the server was done using PUTTY as described in the Linas L (2022) blog. A safe protocol for connecting to a remote server is SSH or Secure Shell. You'll need an SSH client program like PuTTY to create an SSH connection. There is a single-line command to log in using PowerShell or terminal, it is also a technique to get into the SSH server. Another technique is to use the interface of SSH available inside Cpanel.

Once the installation on the server is completed, we need to check if the application is function as same as the local. If not log file must be checked and the problem has to be sorted out. A popular Dashboard hosting platform called Heroku was also used as backup deployment in case of any server failure.



Suicide Dash App Feedback

Name

Email

Phone Number

Message

By Submitting This Form, You Are Providing Express Consent To Receive Commercial Electronic Messages From www.dkit.ie.narayam.net You May Unsubscribe At Any Time.

Suicide Dash App Feedback

Fig 8.5 Feedback section form of the dash app

Appendix A

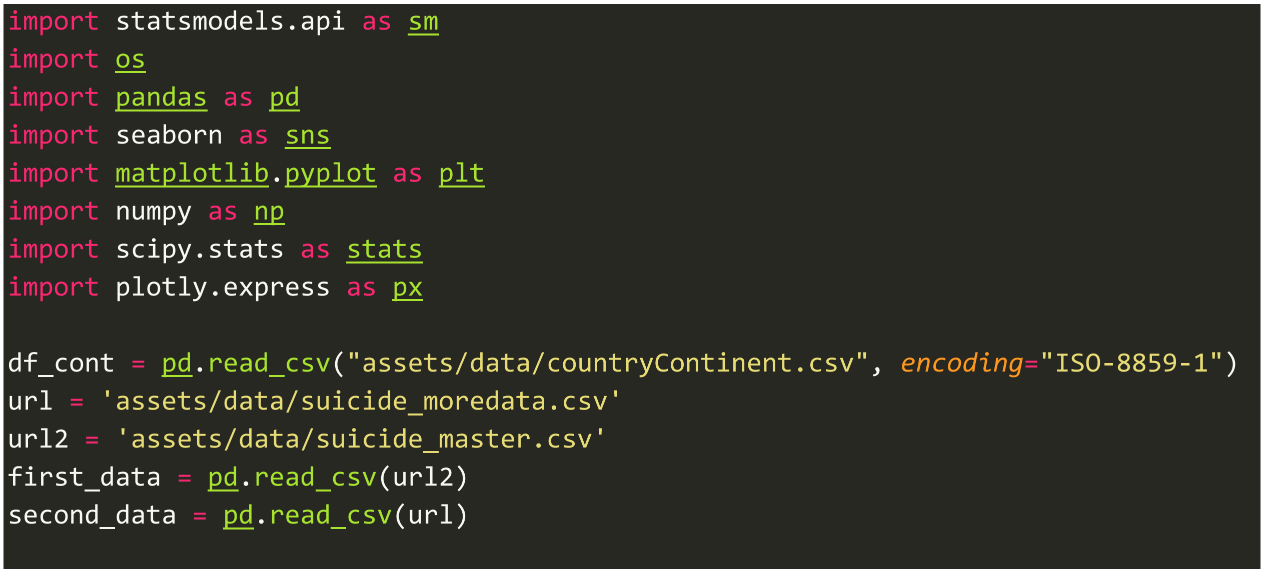
A.1 Data Making in Python

Description: Code to make the datasets to further run the models in server

Authors: Sujil Kumar KM

Available from:

https://github.com/sujilkumarkm/suicide_dash_app_2022/blob/master/data_maker.py



```python
import statsmodels.api as sm
import os
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import numpy as np
import scipy.stats as stats
import plotly.express as px

df_cont = pd.read_csv("assets/data/countryContinent.csv", encoding="ISO-8859-1")
url = 'assets/data/suicide_moredata.csv'
url2 = 'assets/data/suicide_master.csv'
first_data = pd.read_csv(url2)
second_data = pd.read_csv(url)
```

```
first_data.columns = ['country', 'year', 'sex', 'age', 'suicides_no',
'population','suicidesper100k', 'country-year', 'yearlyHDI',
    'GDPpyear', 'GDPpcapita', 'generation']
second_data.columns = ['country', 'year', 'sex', 'age', 'suicides_no',
'population','suicidesper100k', 'country-year', 'yearlyHDI',
    'GDPpyear', 'GDPpcapita', 'generation', 'suicide%', 'Internetusers',
'Expenses', 'employeecompensation','Unemployment', 'Physiciansp1000',
'Legalrights', 'Laborforcetotal','Lifeexpectancy',
'Mobilesubscriptionsp100','Refugees', 'Selfemployed', 'electricityacess',
'secondarycompletion']

second_data.rename( {'GDPpyear':'yearly_gdp' } , axis=1 , inplace = True)
second_data.rename( {'GDPpcapita':'gdp_per_capita' } , axis=1 , inplace = True)
second_data.rename( {'yearlyHDI':'yearly_hdi' } , axis=1 , inplace = True)
second_data.rename( {'suicidesper100k':'sucid_in_hundredk' } , axis=1 , inplace =
True)
second_data.rename( {'suicides_no':'suicides' } , axis=1 , inplace = True)

second_data.columns = map(str.lower, second_data.columns)
# remove special character
second_data.columns = second_data.columns.str.replace(' ', '')

first_data.rename( {'GDPpyear':'yearly_gdp' } , axis=1 , inplace = True)
first_data.rename( {'GDPpcapita':'gdp_per_capita' } , axis=1 , inplace = True)
first_data.rename( {'yearlyHDI':'yearly_hdi' } , axis=1 , inplace = True)
first_data.rename( {'suicidesper100k':'sucid_in_hundredk' } , axis=1 , inplace =
True)
first_data.rename( {'suicides_no':'suicides' } , axis=1 , inplace = True)

first_data.columns = map(str.lower, first_data.columns)
# remove special character
first_data.columns = first_data.columns.str.replace(' ', '')


second_data = pd.merge(second_data, first_data, on =['country', 'year', 'sex',
'age', 'suicides', 'population',
       'sucid_in_hundredk', 'country-year', 'yearly_hdi', 'yearly_gdp',
      'gdp_per_capita', 'generation'] , how = 'left')

second_data = second_data.merge(df_cont[['country', 'continent', 'code_3']])

second_data.rename( {'code_3':'country_code' } , axis=1 , inplace = True)
second_data.rename( {'physiciansp1000':'physician_price' } , axis=1 , inplace =
True)
```

```
second_data.rename( {'mobilesubscriptionsp100':'mobilesubscriptions' } , axis=1 ,
inplace = True)
countries_2 = second_data['country'].unique()

#good sample of the different regions.

countrynames = ['Argentina','Armenia','Australia',    'Austria',
    'Belgium',     'Brazil',     'Bulgaria',     'Canada',     'Chile',     'Colombia'
,    'Croatia',     'Cuba',     'Czech Republic',     'Denmark',
    'Finland',     'France',     'Germany',     'Greece',     'Hungary',     'Iceland'
,    'Ireland', 'Israel','Italy','Japan','Mexico', 'Netherlands','New
Zealand','Norway','Poland', 'Portugal','Romania','Russian Federation','South
Africa', 'Spain','Sweden', 'Switzerland','Thailand',
'Turkmenistan','Ukraine','United Kingdom', 'United States']

# countrynames

df1 = second_data.copy()
final = df1.iloc[np.where(df1.country == countrynames[0])]
for i, x in enumerate(countrynames[1:]):
    final = final.append(df1.iloc[np.where(df1.country == x)])

final = final[final.year >= 1985]
final = final[final.year <= 2016]

final['country'] = final['country'].astype('category')
final['continent'] = final['continent'].astype('category')
final['sex'] = final['sex'].astype('category')
final['generation'] = final['generation'].astype('category')
final['age'] = final['age'].astype('category')

final.drop('yearly_hdi', axis=1, inplace=True)
final.drop('secondarycompletion', axis=1, inplace=True)
final.drop('legalrights', axis=1, inplace=True)

final.internetusers=final.internetusers.fillna(final.internetusers  . min())
final.employeecompensation=final.employeecompensation.fillna(final.employeecompen
sation.mean())
final.electricityacess=final.electricityacess.fillna(final.electricityacess.mean(
))
final.refugees=final.refugees.fillna(final.refugees.mean())
final.expenses=final.expenses.fillna(final.expenses.mean())
final.physician_price=final.physician_price.fillna(final.physician_price.mean())
```

```
final['internetusers'] = final['internetusers'].replace(r'^\s*$', np.nan,
regex=True)
final['unemployment'] = final['unemployment'].replace(r'^\s*$', np.nan,
regex=True)
final['physician_price'] = final['physician_price'].replace(r'^\s*$', np.nan,
regex=True)
final['internetusers'] = final['internetusers'].replace(r'^\s*$', np.nan,
regex=True)
final['laborforcetotal'] = final['laborforcetotal'].replace(r'^\s*$', np.nan,
regex=True)
final['selfemployed'] = final['selfemployed'].replace(r'^\s*$', np.nan,
regex=True)
final['electricityacess'] = final['electricityacess'].replace(r'^\s*$', np.nan,
regex=True)
final['lifeexpectancy'] = final['lifeexpectancy'].replace(r'^\s*$', np.nan,
regex=True)
final['mobilesubscription'] = final['mobilesubscriptions'].replace(r'^\s*$',
np.nan, regex=True)
final['refugees'] = final['refugees'].replace(r'^\s*$', np.nan, regex=True)
final['expenses'] = final['expenses'].replace(r'^\s*$', np.nan, regex=True)
final['employeecompensation'] = final['employeecompensation'].replace(r'^\s*$',
np.nan, regex=True)
final['physician_price'] = final['physician_price'].replace(r'^\s*$', np.nan,
regex=True)


final.loc[ final['internetusers'] == 0 | np.isnan(final['internetusers']),
'internetusers' ] = final['internetusers'].mean()
final.loc[ final['unemployment'] == 0 | np.isnan(final['unemployment']),
'unemployment' ] = final['unemployment'].mean()
final.loc[ final['physician_price'] == 0 | np.isnan(final['physician_price']),
'physician_price' ] = final['physician_price'].min()
final.loc[ final['laborforcetotal'] == 0 | np.isnan(final['laborforcetotal']),
'laborforcetotal' ] = final['laborforcetotal'].mean()
final.loc[ final['selfemployed'] == 0 | np.isnan(final['selfemployed']),
'selfemployed' ] = final['selfemployed'].mean()
final.loc[ final['electricityacess'] == 0 | np.isnan(final['electricityacess']),
'electricityacess' ] = final['electricityacess'].mean()
final.loc[ final['lifeexpectancy'] == 0 | np.isnan(final['lifeexpectancy']),
'lifeexpectancy' ] = final['lifeexpectancy'].mean()
final.loc[ final['mobilesubscriptions'] == 0 |
np.isnan(final['mobilesubscriptions']), 'mobilesubscriptions' ] =
final['mobilesubscriptions'].mean()
final.loc[ final['refugees'] == 0 | np.isnan(final['refugees']), 'refugees' ] =
final['refugees'].mean()
```

```
final.loc[ final['expenses'] == 0 | np.isnan(final['expenses']), 'expenses' ] =
final['expenses'].mean()
final.loc[ final['employeecompensation'] == 0 |
np.isnan(final['employeecompensation']), 'employeecompensation' ] =
final['employeecompensation'].mean()
final.loc[ final['physician_price'] == 0 | np.isnan(final['physician_price']),
'physician_price' ] = final['physician_price'].mean()

final.loc[:, 'expenses':'refugees'] = final.loc[:,
'expenses':'refugees'].fillna(final['employeecompensation'].mean())



final.to_csv('assets/processed_data/output.csv',mode = 'w', index=False)
print("File saved successfully!!!")
# outputting data to run models in live server
```

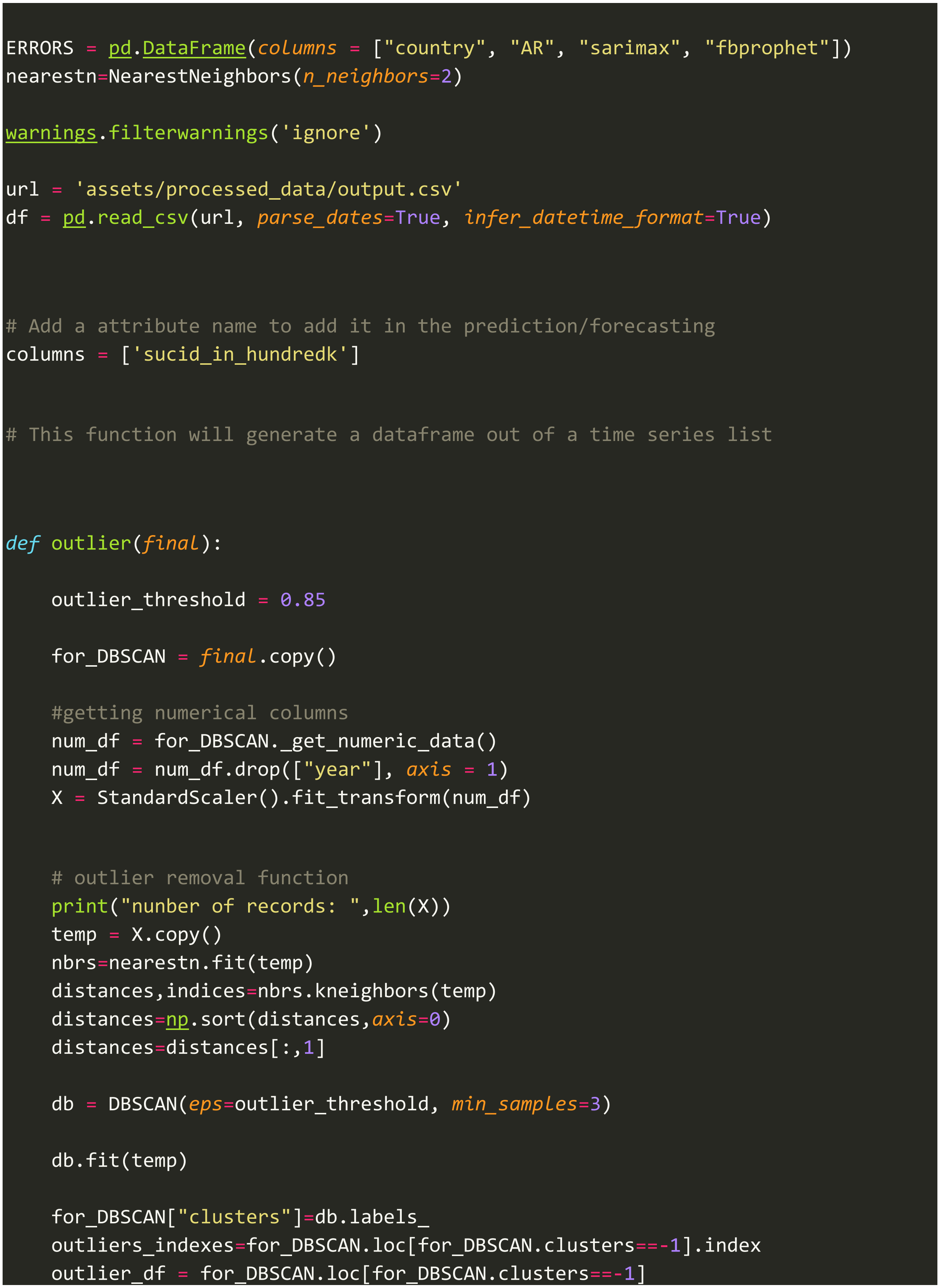
A.2 Modeling and Outlier Detection

Description: Three different Model including FB Prophet, Custom AR and SARIMAX, Periodic Outlier Detection

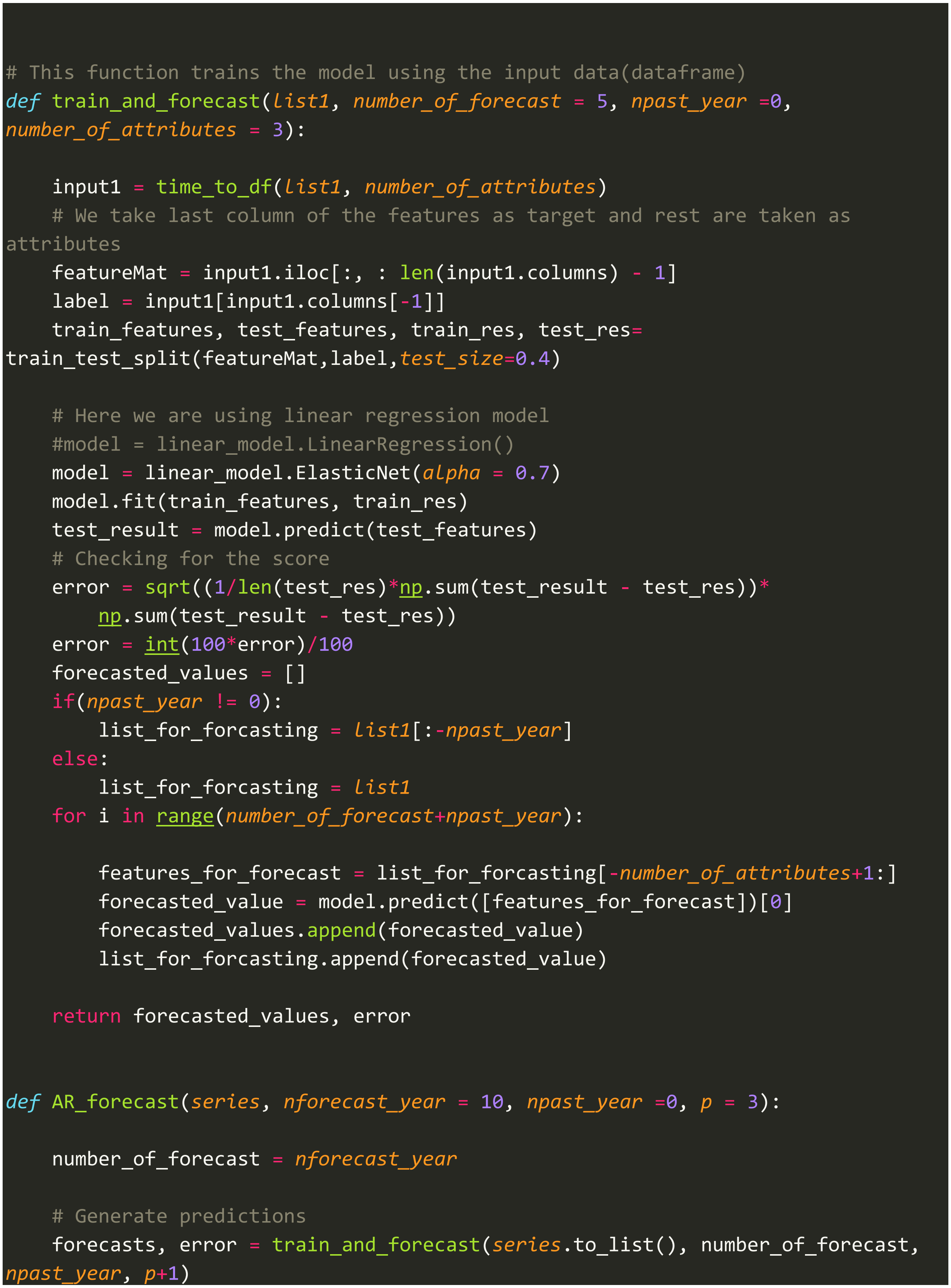
Authors: Sujil Kumar K.M

Available from: https://github.com/sujilkumarkm/suicide_dash_app_2022/blob/master/data_modelling.py

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from math import sqrt
from pandas import read_csv
from sklearn import linear_model
from sklearn.model_selection import train_test_split
import datetime
from prophet import Prophet
from sklearn.metrics import mean_squared_error
import dateutil.parser # for handling the conversion of datetime formats
from datetime import timedelta # for operating the datetime objects
from statsmodels.tsa. statespace.sarimax import SARIMAX
from statsmodels.tools.eval_measures import rmse
from tqdm import tqdm
from os import system, name
import warnings
from sklearn.cluster import DBSCAN
from sklearn.neighbors import NearestNeighbors
from sklearn.preprocessing import StandardScaler
```

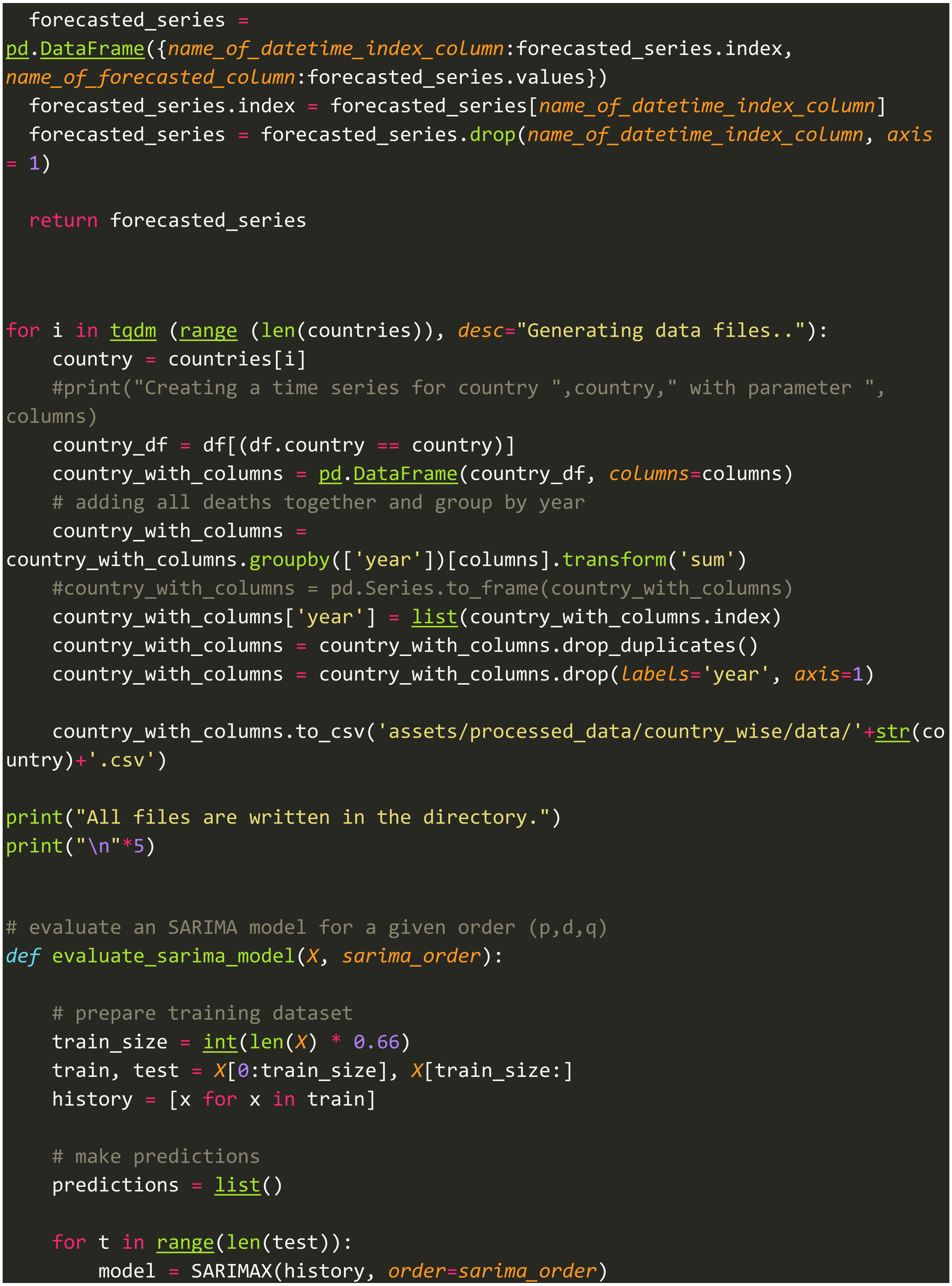


```
ERRORS = pd.DataFrame(columns = ["country", "AR", "sarimax", "fbprophet"])
nearestn=NearestNeighbors(n_neighbors=2)

warnings.filterwarnings('ignore')

url = 'assets/processed_data/output.csv'
df = pd.read_csv(url, parse_dates=True, infer_datetime_format=True)



# Add a attribute name to add it in the prediction/forecasting
columns = ['sucid_in_hundredk']


# This function will generate a dataframe out of a time series list



def outlier(final):

    outlier_threshold = 0.85

    for_DBSCAN = final.copy()

    #getting numerical columns
    num_df = for_DBSCAN._get_numeric_data()
    num_df = num_df.drop(["year"], axis = 1)
    X = StandardScaler().fit_transform(num_df)


    # outlier removal function
    print("nunber of records: ",len(X))
    temp = X.copy()
    nbrs=nearestn.fit(temp)
    distances,indices=nbrs.kneighbors(temp)
    distances=np.sort(distances,axis=0)
    distances=distances[:,1]

    db = DBSCAN(eps=outlier_threshold, min_samples=3)

    db.fit(temp)

    for_DBSCAN["clusters"]=db.labels_
    outliers_indexes=for_DBSCAN.loc[for_DBSCAN.clusters==-1].index
    outlier_df = for_DBSCAN.loc[for_DBSCAN.clusters==-1]
    print("Total ",len(outliers_indexes)," are outliers")

    core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
    core_samples_mask[db.core_sample_indices_] = True
    labels = db.labels_

    # Number of clusters in labels, ignoring noise if present.
    n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
    n_noise_ = list(labels).count(-1)

    print("Estimated number of clusters: %d" % n_clusters_)
    print("Estimated number of noise points: %d" % n_noise_)

    for_DBSCAN=for_DBSCAN.drop(outliers_indexes,axis=0)
    for_DBSCAN=for_DBSCAN.drop("clusters",axis=1)

    print("Before removing outliers, total number of records: ", len(final))
    final=final.drop(outliers_indexes,axis=0)

    print("After removing outliers, total number of records: ", len(final))
    print("#"*40)

    return final, outlier_df

df, outlier_df = outlier(df)
df.index = df.year
df = df.sort_index(axis=0, level=None, ascending=True, inplace=False,
kind='quicksort', na_position='last', sort_remaining=True, ignore_index=False,
key=None)

outlier_df.to_csv('assets/processed_data/outliers.csv')

countries=df['country'].unique()

def time_to_df(list1, number_of_attributes = 3):
    df = pd.DataFrame(columns=range(number_of_attributes))

    for i in range(len(list1)-number_of_attributes+1):
        record = []
        for j in range(i,i+number_of_attributes):
            record.append(list1[j])
        df.loc[len(df.index)] = record


    return df
```

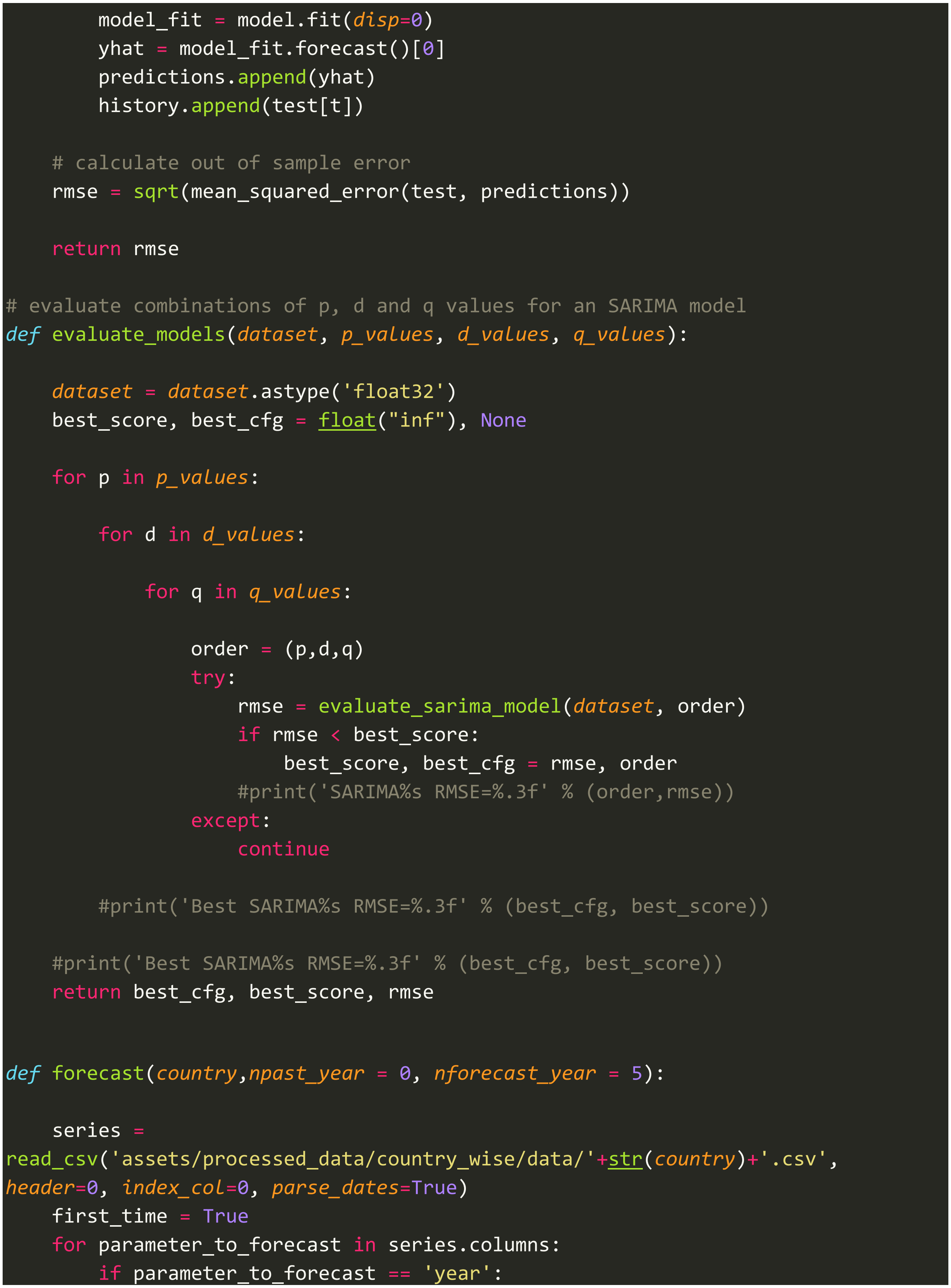


```
# This function trains the model using the input data(dataframe)
def train_and_forecast(list1, number_of_forecast = 5, npast_year =0,
number_of_attributes = 3):

    input1 = time_to_df(list1, number_of_attributes)
    # We take last column of the features as target and rest are taken as
attributes
    featureMat = input1.iloc[:, : len(input1.columns) - 1]
    label = input1[input1.columns[-1]]
    train_features, test_features, train_res, test_res=
train_test_split(featureMat,label,test_size=0.4)

    # Here we are using linear regression model
    #model = linear_model.LinearRegression()
    model = linear_model.ElasticNet(alpha = 0.7)
    model.fit(train_features, train_res)
    test_result = model.predict(test_features)
    # Checking for the score
    error = sqrt((1/len(test_res)*np.sum(test_result - test_res))*
        np.sum(test_result - test_res))
    error = int(100*error)/100
    forecasted_values = []
    if(npast_year != 0):
        list_for_forcasting = list1[:-npast_year]
    else:
        list_for_forcasting = list1
    for i in range(number_of_forecast+npast_year):

        features_for_forecast = list_for_forcasting[-number_of_attributes+1:]
        forecasted_value = model.predict([features_for_forecast])[0]
        forecasted_values.append(forecasted_value)
        list_for_forcasting.append(forecasted_value)

    return forecasted_values, error


def AR_forecast(series, nforecast_year = 10, npast_year =0, p = 3):

    number_of_forecast = nforecast_year

    # Generate predictions
    forecasts, error = train_and_forecast(series.to_list(), number_of_forecast,
npast_year, p+1)
```

```
    forecasts_ser = pd.Series(forecasts, copy=False)
    to_plot=forecasted_series_to_df(series, forecasts_ser, npast_year,
str(series.name), "Date")

    return to_plot, series, error

# function to check if the year is leap year or not
def is_leap_year(year):

    if (year%4) == 0:
        if (year%100) == 0:
            if (year%400) == 0:
                return True
            else:
                return False
        else:
             return True
    else:
        return False

def clear():
    if name == 'nt':
        _ = system('cls')
    else:
        _ = system('clear')

def forecasted_series_to_df(series, forecasted_series_, npast_year,
name_of_forecasted_column, name_of_datetime_index_column):
  forecasted_series = forecasted_series_.copy()
  y = 0
  index_for_forcaste = []
  index_for_forcaste.append(series.index[-npast_year-1])
  for i in range(len(forecasted_series)-1):
    y = y+1
    date_temp = index_for_forcaste[-1]
    if(is_leap_year(date_temp.year)):
      date_temp = date_temp + timedelta(days = 366)
    else:
      date_temp = date_temp + timedelta(days = 365)
    index_for_forcaste.append(date_temp)


  forecasted_series.index = pd.to_datetime(index_for_forcaste)
```



```
  forecasted_series =
pd.DataFrame({name_of_datetime_index_column:forecasted_series.index,
name_of_forecasted_column:forecasted_series.values})
  forecasted_series.index = forecasted_series[name_of_datetime_index_column]
  forecasted_series = forecasted_series.drop(name_of_datetime_index_column, axis
= 1)

  return forecasted_series



for i in tqdm (range (len(countries)), desc="Generating data files.."):
    country = countries[i]
    #print("Creating a time series for country ",country," with parameter ",
columns)
    country_df = df[(df.country == country)]
    country_with_columns = pd.DataFrame(country_df, columns=columns)
    # adding all deaths together and group by year
    country_with_columns =
country_with_columns.groupby(['year'])[columns].transform('sum')
    #country_with_columns = pd.Series.to_frame(country_with_columns)
    country_with_columns['year'] = list(country_with_columns.index)
    country_with_columns = country_with_columns.drop_duplicates()
    country_with_columns = country_with_columns.drop(labels='year', axis=1)

    country_with_columns.to_csv('assets/processed_data/country_wise/data/'+str(co
untry)+'.csv')

print("All files are written in the directory.")
print("\n"*5)


# evaluate an SARIMA model for a given order (p,d,q)
def evaluate_sarima_model(X, sarima_order):

    # prepare training dataset
    train_size = int(len(X) * 0.66)
    train, test = X[0:train_size], X[train_size:]
    history = [x for x in train]

    # make predictions
    predictions = list()

    for t in range(len(test)):
        model = SARIMAX(history, order=sarima_order)
```



```
        model_fit = model.fit(disp=0)
        yhat = model_fit.forecast()[0]
        predictions.append(yhat)
        history.append(test[t])

    # calculate out of sample error
    rmse = sqrt(mean_squared_error(test, predictions))

    return rmse

# evaluate combinations of p, d and q values for an SARIMA model
def evaluate_models(dataset, p_values, d_values, q_values):

    dataset = dataset.astype('float32')
    best_score, best_cfg = float("inf"), None

    for p in p_values:

        for d in d_values:

            for q in q_values:

                order = (p,d,q)
                try:
                    rmse = evaluate_sarima_model(dataset, order)
                    if rmse < best_score:
                        best_score, best_cfg = rmse, order
                    #print('SARIMA%s RMSE=%.3f' % (order,rmse))
                except:
                    continue

        #print('Best SARIMA%s RMSE=%.3f' % (best_cfg, best_score))

    #print('Best SARIMA%s RMSE=%.3f' % (best_cfg, best_score))
    return best_cfg, best_score, rmse


def forecast(country,npast_year = 0, nforecast_year = 5):

    series =
read_csv('assets/processed_data/country_wise/data/'+str(country)+'.csv',
header=0, index_col=0, parse_dates=True)
    first_time = True
    for parameter_to_forecast in series.columns:
        if parameter_to_forecast == 'year':
            pass
        else:
            # evaluate parameters
            p_values = [0, 1, 2, 4, 6]
            d_values = range(0, 3)
            q_values = range(0, 3)

            tdf = series[parameter_to_forecast].copy()
            tdf.index = series.index
            tdf = tdf.squeeze()

            # selecting best model using grid search
            best_cfg, best_score, error_sarimax = evaluate_models(tdf.values,
p_values, d_values, q_values)
            # Instantiate the model

            model = SARIMAX(series[parameter_to_forecast], order=best_cfg)

            # Fit the model
            results = model.fit()

            # Generate predictions
            forecasts = results.get_prediction(start=len(series)-npast_year,end =
len(series)+nforecast_year-1)
            forecasted_1, actual, error_AR =
AR_forecast(series[parameter_to_forecast],
                nforecast_year = nforecast_year,npast_year = npast_year, p = 5)
            forcasted_final = [actual.to_list()[-1]]
            forcasted_final.extend(forecasted_1[parameter_to_forecast].to_list())

            data_fbp = series.copy()
            data_fbp["year"] = data_fbp.index
            data_fbp.columns = ['y','ds']
            data_fbp['ds'] = pd.to_datetime(data_fbp['ds'])
            train = data_fbp.iloc[:len(data_fbp)-int(len(data_fbp)*0.25)]
            test = data_fbp.iloc[len(data_fbp)-int(len(data_fbp)*0.25)+1:]
            m = Prophet(interval_width = 0.80)
            m.fit(data_fbp)
            future = m.make_future_dataframe(periods=15, freq = "Y",
include_history = "False")
            forecast = m.predict(future)
            res = forecast[['ds', 'yhat', 'yhat_lower', 'yhat_upper']]
            forcecast_fbf = [actual.to_list()[-1]]
            temp_ls = forecast['yhat'].to_list()[len(data_fbp):]
            forcecast_fbf.extend(temp_ls)
            forcecast_fbf = pd.Series(forcecast_fbf, copy=False)
            to_plot_fbf=forecasted_series_to_df(series[parameter_to_forecast],
forcecast_fbf, npast_year, str(parameter_to_forecast), "year")

            # Model evaluation for prophet
            predictions = forecast.iloc[-len(test):]['yhat']
            #print("Root Mean Squared Error between actual and  predicted values:
",rmse(predictions,test['y']), "-->
",int(rmse(predictions,test['y'])*10000/test['y'].mean())/100," %")
            error_fbprophet = rmse(predictions,test['y'])
            mean_forecast_sarima = forecasts.predicted_mean

            forcasted_final_sarima = [actual.to_list()[-1]]
            forcasted_final_sarima.extend(mean_forecast_sarima.to_list())
            forcasted_final_sarima = pd.Series(forcasted_final_sarima,
copy=False)

            print("Forecasting for country: ", country)

            to_plot=forecasted_series_to_df(series[parameter_to_forecast],
forcasted_final_sarima, npast_year, str(parameter_to_forecast), "year")
            to_plot[parameter_to_forecast+"_sarimax"] =
to_plot[parameter_to_forecast]
            to_plot[parameter_to_forecast+"_AR"] = forcasted_final
            to_plot=to_plot.drop([parameter_to_forecast], axis = 1)
            to_plot[parameter_to_forecast+"_fbprophet"] =
to_plot_fbf[parameter_to_forecast]

            to_plot[parameter_to_forecast+"_sarimax"] =
to_plot[parameter_to_forecast+"_sarimax"].apply(lambda x: int(x*100)/100)
            to_plot[parameter_to_forecast+"_AR"] =
to_plot[parameter_to_forecast+"_AR"].apply(lambda x: int(x*100)/100)
            to_plot[parameter_to_forecast+"_fbprophet"] =
to_plot[parameter_to_forecast+"_fbprophet"].apply(lambda x: int(x*100)/100)

            if first_time:
                first_time = False
                temp = to_plot.copy()
            else:
                temp=pd.merge(to_plot, temp, on = "year", how = 'right')

            ERRORS.loc[len(ERRORS)] = [country, error_AR,error_sarimax,
error_fbprophet]
```

```
    # writting forecastes to the hard-disk
    temp.to_csv('assets/processed_data/country_wise/forecasted/'+str(country)+'.c
sv')

    return temp, series
clear()
for i in tqdm (range (len(countries)), desc="Generating forecasted data
files.."):
    country = countries[i]
    to_plot, series = forecast(country = country, npast_year = 0, nforecast_year
= 15)
    ERRORS.to_csv('assets/processed_data/error.csv')
    print("Error file written successfully.")
    clear()
clear()
print("Forecasted files are ready to serve the dash boarded.")
print("Error file written successfully.")
```

5.0 Bibliography

10 Reasons Why Laravel Is The Best PHP Framework For 2022. (n.d.). Retrieved August 25, 2022, from https://www.clariontech.com/blog/10-reasons-why-laravel-is-the-best-php-framework-for-2019

Aborujilah, A., Adamu, J., Shariff, S. M., & Long, Z. A. (2022). Descriptive Analysis of Built-in Security Features in Web Development Frameworks. *Proceedings of the 2022 16th International Conference on Ubiquitous Information Management and Communication, IMCOM 2022*. https://doi.org/10.1109/IMCOM53663.2022.9721750

Ajitesh Kumar. (2022, May 28). *Purpose of Dashboard: Advantages & Disadvantages - Data Analytics*. https://vitalflux.com/dashboard-purpose-advantages-disadvantages/

Alexander Blaufuss. (2020). *How To Build A Dashboard In Python – Plotly Dash Step-by-Step Tutorial*. https://www.statworx.com/en/content-hub/blog/how-to-build-a-dashboard-in-python-plotly-dash-step-by-step-tutorial/

Aljaaf, A. J., Al-Jumeily, D., Hussain, A. J., Dawson, T., Fergus, P., & Al-Jumaily, M. (2015). Predicting the likelihood of heart failure with a multi level risk assessment using decision tree. *2015 3rd International Conference on Technological Advances in Electrical, Electronics and Computer Engineering, TAEECE 2015*, 101–106. https://doi.org/10.1109/TAEECE.2015.7113608

Almasi, K., Belso, N., Kapur, N., Webb, R., Cooper, J., Hadley, S., Kerfoot, M., Dunn, G., Sotonyi, P., Rihmer, Z., & Appleby, L. (2009). Risk factors for suicide in Hungary: A case-control study. *BMC Psychiatry*, *9*(1), 1–9. https://doi.org/10.1186/1471-244X-9-45/TABLES/4

Anna Politkovskaya. (2004). *Politkovskaya A.Putin's Russia.*

Brendan Artley. (2022). *Time Series Forecasting with ARIMA , SARIMA and SARIMAX | by Brendan Artley | Towards Data Science*. https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6

Chatfield, C. (2000). *Time-Series Forecasting*. https://doi.org/10.1201/9781420036206

Copenhagen: World Health Organization. (2005). *WHO Regional Office for Europe Mental health*.

CSO statistical release. (2014). *Suicide Statistics 2011 - CSO - Central Statistics Office*. https://www.cso.ie/en/releasesandpublications/er/ss/suicidestatistics2011/

David Hurst. (2021). *Why being wealthy can make mental health problems worse - Tikvah Lake Florida*. https://www.tikvahlake.com/blog/why-being-wealthy-can-make-mental-health-problems-worse/

Dylan Castillo. (2022). *Develop Data Visualization Interfaces in Python With Dash – Real Python*. https://realpython.com/python-dash/

Elias Dabbas. (2018). *Migration and Population Density - WorldBank Data Dashboard*. https://www.dashboardom.com/migration-population

Elias Dabbas. (2019). *Build a Ploty Dash App - Poverty Data Dashboard*. https://povertydata.org/

Exploratory Data Analysis (EDA) using Python and Jupyter Notebooks - YouTube. (n.d.). IGeek. Retrieved August 27, 2022, from https://www.youtube.com/watch?v=iZ2MwVWKwr4

Geckoboard. (2022). *Sales YTD*. https://share.geckoboard.com/dashboards/JPVRGXEPTTUBIX4Y?_ga=2.223387931.1481208763.1662722437-1665380050.1662722437

Havasi, B., Mágori, K., Tóth, A., & Kiss, L. (2005). Fatal suicide cases from 1991 to 2000 in Szeged, Hungary. *Forensic Science International*, *147*(SUPPL.), S25–S28. https://doi.org/10.1016/J.FORSCIINT.2004.09.092

Hosting a Dash app on a VPS.. As data scientists, we generally seem… | by Umar Khan | Analytics Vidhya | Medium. (n.d.). Retrieved August 25, 2022, from https://medium.com/analytics-vidhya/hosting-a-dash-app-on-a-vps-7cb56fa310b9

How to Secure SSH | cPanel & WHM Documentation. (n.d.). Retrieved August 25, 2022, from https://docs.cpanel.net/knowledge-base/security/how-to-secure-ssh/

Huang, Z., & Chalabi, Z. S. (1995). Use of time-series analysis to model and forecast wind speed. *Journal of Wind Engineering and Industrial Aerodynamics*, *56*(2–3), 311–322. https://doi.org/10.1016/0167-6105(94)00093-S

James Watkins. (2017). *The Story Behind Russia's Male Suicide Problem | OZY*. https://www.ozy.com/around-the-world/the-story-behind-russias-male-suicide-problem/76845/

Joel Gunter. (2022). *How suicide became the hidden toll of the war in Ukraine - BBC News*. https://www.bbc.com/news/world-europe-60318298

Johns Hopkins University. (2022). *COVID-19 Map - Johns Hopkins Coronavirus Resource Center*. https://coronavirus.jhu.edu/map.html

Khanyi Mlaba. (2022, February 1). *Water Scarcity in Africa: Everything You Need to Know*. Globalcitizen.Org. https://www.globalcitizen.org/en/content/water-scarcity-in-africa-explainer-what-to-know/

Kumar Jha, B., & Pande, S. (2021). Time Series Forecasting Model for Supermarket Sales using FB-Prophet. *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, 547–554. https://doi.org/10.1109/ICCMC51019.2021.9418033

Linas L. (2022). *How to Use PuTTY SSH Client on Windows, Mac and Linux*. https://www.hostinger.com/tutorials/how-to-use-putty-ssh

Markus Schmitt. (2022). *Data dashboarding tools | Streamlit v.s. Dash v.s. Shiny vs. Voila vs. Flask vs. Jupyter*. https://www.datarevenue.com/en-blog/data-dashboarding-streamlit-vs-dash-vs-shiny-vs-voila

McDermott, R. N. (2016). The Return of the Cold War: Ukraine, the West and Russia. In *The Return of the Cold War*. Routledge. https://doi.org/10.4324/9781315684567-5

National Office for Suicide Prevention, H. (n.d.). *National Education and Training Plan, January 2022*. Retrieved August 28, 2022, from www.nosp.ie

Nordstrom, D. L. (2007). Ukraine set to act on high suicide burden. *Injury Prevention*, *13*(4), 224. https://doi.org/10.1136/IP.2007.015768

Patowary, A. N., Pratim Barman, M., & Rao Gadde, S. (2018). *Accidental Deaths in India : Forecasting with ARIMA Model Control charts View project Bayesian Time series Analysis View project*. https://www.researchgate.net/publication/324919218

Pecoraro, C. J., John, C., Reviewers, P., Coyle, K., Lim, J., & Maity, C. (2015). *Mastering Laravel Develop robust modern web-based software applications and RESTful APIs with Laravel, one of the hottest PHP frameworks Mastering Laravel Credits*. www.packtpub.com

pmdarima: ARIMA estimators for Python — pmdarima 2.0.1 documentation. (n.d.). Retrieved August 27, 2022, from http://alkaline-ml.com/pmdarima/

Stan - Stan. (n.d.). Retrieved August 27, 2022, from https://mc-stan.org/

Steven Gruzd. (2022). *Africa: Key issues to track in 2022*. https://www.africaportal.org/features/africa-key-issues-track-2022/

Suicide and suicidal thoughts - Diagnosis and treatment - Mayo Clinic. (n.d.). Retrieved August 28, 2022, from https://www.mayoclinic.org/diseases-conditions/suicide/diagnosis-treatment/drc-20378054

Suicide Statistics 2011 - CSO - Central Statistics Office. (n.d.). Retrieved August 28, 2022, from https://www.cso.ie/en/releasesandpublications/er/ss/suicidestatistics2011/

The Economist. (2018). *Why the global suicide rate is falling | The Economist*. https://www.economist.com/the-economist-explains/2018/11/30/why-the-global-suicide-rate-is-falling

Time Series Analysis: Definition, Types & Techniques | Tableau. (n.d.). Retrieved August 24, 2022, from https://www.tableau.com/learn/articles/time-series-analysis

UN. (2015). *World Economic Situation and Prospects 2015 | Department of Economic and Social Affairs*. https://www.un.org/development/desa/dpad/publication/world-economic-situation-and-prospects-2015/

Vera Shao. (2020). *Forecasting with a Time Series Model using Python: Part One | Bounteous*. https://www.bounteous.com/insights/2020/09/15/forecasting-time-series-model-using-python-part-one/

Zetzsche, T., Bobes, J., de La Fuente, J. M., Pogarell, O., Norra, C., Schmidtke, A., Wasserman, D., Löhr, C., & Rihmer, Z. (2007). Changing suicide rates in western and central Europe. *European Psychiatry*, *22*(S1), S35–S35. https://doi.org/10.1016/J.EURPSY.2007.01.140

(3) 1/4: What is Streamlit - YouTube. Available from: https://www.youtube.com/watch?v=R2nr1uZ8ffc [accessed 4 June 2022].

Airiti Library_Comparative+Study+of+Artificial+Neural+Network+and+ARIMA+Models+in+Predicting+Exchange+Rate. Available from: https://www.airitilibrary.com/Publication/alDetailedMesh?docid=20407467-201211-201512080011-201512080011-4397-4403 [accessed 8 April 2022].

Akbari, Z. and Unland, R. (2016). Automated determination of the input parameter of DBSCAN based on outlier detection. *IFIP Advances in Information and Communication Technology* [online], 475, pp.280–291. Available from: https://link.springer.com/chapter/10.1007/978-3-319-44944-9_24 [accessed 21 August 2022].

Bellman, V. and Namdev, V. (2022). Suicidality Among Men in Russia: A Review of Recent Epidemiological Data. *Cureus* [online], 14(3). Available from: https://www.cureus.com/articles/88128-suicidality-among-men-in-russia-a-review-of-recent-epidemiological-data [accessed 8 June 2022].

Bogod, D. (2004). The Nazi Hypothermia Experiments: Forbidden Data?. *Anaesthesia* [online], 59(12), pp.1155–1156.

Brunello, A., Marzano, E., Montanari, A. and Sciavicco, G. (2019). J48SS: A Novel Decision Tree Approach for the Handling of Sequential and Time Series Data. *Computers 2019, Vol. 8, Page 21* [online], 8(1), p.21. Available from: https://www.mdpi.com/2073-431X/8/1/21/htm [accessed 7 June 2022].

Build a Ploty Dash App - Poverty Data Dashboard. Available from: https://povertydata.org/ [accessed 23 August 2022].

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning* [online], 20(3), pp.273–297.

COVID-19 Map - Johns Hopkins Coronavirus Resource Center. Available from: https://coronavirus.jhu.edu/map.html [accessed 11 June 2022].

Facebook Files: 5 things leaked documents reveal - BBC News. Available from: https://www.bbc.com/news/technology-58678332 [accessed 12 June 2022].

Filho, D.M. and Valk, M. (2020). Dynamic VAR model-based control charts for batch process monitoring. *European Journal of Operational Research* [online], 285(1), pp.296–305.

Fujita, A., Sato, J.R., Garay-Malpartida, H.M., Yamaguchi, R., Miyano, S., Sogayar, M.C. and Ferreira, C.E. (2007). Modeling gene expression regulatory networks with the sparse vector autoregressive model. *BMC Systems Biology* [online], 1.

How suicide became the hidden toll of the war in Ukraine - BBC News. Available from: https://www.bbc.com/news/world-europe-60318298 [accessed 8 June 2022].

Huang, M.W., Chen, C.W., Lin, W.C., Ke, S.W. and Tsai, C.F. (2017). SVM and SVM ensembles in breast cancer prediction. *PLoS ONE*, 12(1).

Human Radiation Experiments | Atomic Heritage Foundation. Available from: https://www.atomicheritage.org/history/human-radiation-experiments [accessed 15 April 2022].

John, A., Glendenning, A.C., Marchant, A., Montgomery, P., Stewart, A., Wood, S., Lloyd, K. and Hawton, K. (2018). Self-Harm, Suicidal Behaviours, and Cyberbullying in Children and Young People: Systematic Review. *J Med Internet Res 2018;20(4):e129 https://www.jmir.org/2018/4/e129* [online], 20(4), p.e9044. Available from: https://www.jmir.org/2018/4/e129 [accessed 9 June 2022].

John, A., Okolie, C., Eyles, E., Webb, R.T., Schmidt, L., McGuiness, L.A., Olorisade, B.K., Arensman, E., Hawton, K., Kapur, N., Moran, P., O'Connor, R.C., O'Neill, S., Higgins, J.P.T. and Gunnell, D. (2020). The impact of the COVID-19 pandemic on self-harm and suicidal behaviour: a living systematic review. *F1000Research 2020 9:1097* [online], 9, p.1097. Available from: https://f1000research.com/articles/9-1097 [accessed 7 June 2022].

Kumar, N. and Susan, S. (2020a). COVID-19 Pandemic Prediction using Time Series Forecasting Models. *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020* [online], 1 July 2020.

Kumar, N. and Susan, S. (2020b). COVID-19 Pandemic Prediction using Time Series Forecasting Models. *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020* [online], 1 July 2020.

Lee, K.C. and Oh, S.B. (1996). An intelligent approach to time series identification by a neural network-driven decision tree classifier. *Decision Support Systems* [online], 17(3), pp.183–197.

Mochahost Review 2022: Mocha Host Details, Pricing & Features | Sitechecker. Available from: https://sitechecker.pro/web-hosting/mochahost.com/ [accessed 7 June 2022].

Naim, N.F., Mohd Yassin, A.I., Zamri, W.M.A.W. and Sarnin, S.S. (2011). MySQL database for storage of fingerprint data. *Proceedings - 2011 UKSim 13th International Conference on Modelling and Simulation, UKSim 2011* [online], 2011, pp.293–298.

Qi, F., Xu, Z., Zhang, H., Wang, R., Wang, Y., Jia, X., Lin, P., Geng, M., Huang, Y., Li, S. and Yang, J. (2021). Predicting the mortality of smoking attributable to cancer in Qingdao, China: A time-series analysis. *PLOS ONE* [online], 16(1), p.e0245769. Available from: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0245769 [accessed 8 April 2022].

Singh, V., Poonia, R.C., Kumar, S., Dass, P., Agarwal, P., Bhatnagar, V. and Raja, L. (2020). Prediction of COVID-19 corona virus pandemic based on time series data using support vector machine. *https://doi.org/10.1080/09720529.2020.1784535* [online], 23(8), pp.1583–1597. Available from: https://www.tandfonline.com/doi/abs/10.1080/09720529.2020.1784535 [accessed 8 June 2022].

Study: Benefits of Electric Cars Add Up—in the Billions! | NRDC. Available from: https://www.nrdc.org/experts/luke-tonachel/study-benefits-electric-cars-add-billions [accessed 15 April 2022].

Suicide Statistics 2011 - CSO - Central Statistics Office. Available from: https://www.cso.ie/en/releasesandpublications/er/ss/suicidestatistics2011/ [accessed 7 June 2022].

Tang, L., Pan, H. and Yao, Y. (2018). K-nearest neighbor regression with principal component analysis for financial time series prediction. *ACM International Conference Proceeding Series* [online], 12 March 2018, pp.127–131.

Värnik, P. (2012). Suicide in the World. *International Journal of Environmental Research and Public Health* [online], 9(3), pp.760–771.

Vector Autoregressive Models for Multivariate Time Series. (2006). *Modeling Financial Time Series with S-PLUS®* [online], 9 October 2006, pp.385–429. Available from: https://link.springer.com/chapter/10.1007/978-0-387-32348-0_11 [accessed 4 June 2022].

When Quaker Oats Fed Children Radioactive Oatmeal | by Calin Aneculaesei | History of Yesterday. Available from: https://historyofyesterday.com/when-quaker-oats-fed-children-radioactive-oatmeal-5e06faf3ce4d [accessed 9 June 2022].

Włodarczyk, T., Płotka, S., Szczepański, T., Rokita, P., Sochacki-Wójcicka, N., Wójcicki, J., Lipa, M. and Trzciński, T. (2021). Machine learning methods for preterm birth prediction: A review. *Electronics (Switzerland)*, 10(5).

Zetzsche, T., Bobes, J., de La Fuente, J.M., Pogarell, O., Norra, C., Schmidtke, A., Wasserman, D., Löhr, C. and Rihmer, Z. (2007). Changing suicide rates in western and central Europe. *European Psychiatry* [online], 22(S1), pp.S35–S35. Available from: https://www.cambridge.org/core/journals/european-psychiatry/article/changing-suicide-rates-in-western-and-central-europe/1B2C943626D6D31150E00FCD3CECFDA9 [accessed 11 June 2022].