**Visualizing the Impact of Covid-19 in Indonesia Using Tableau and Python**

**COM 618(Data Science)**

ASSESSMENT REPORT

BY

DAVID AGBOLADE

STUDENT NO.: Q14872382

BSC(Hons) SOFTWARE ENGINEERING

**INTRODUCTION**

Indonesia has had significant effects from Covid-19 due to the virus' quick national spread (World Health Organization, 2020). In order to make informed decisions and put into practise efficient tactics to stop its spread, it is essential to understand the course of the pandemic in Indonesia (Ministry of Health, Republic of Indonesia, 2020).

One way to visualize and understand the data on Covid-19 in Indonesia is through the use of data visualization tools, such as Tableau and Python. These tools allow us to create interactive dashboards and graphs that enable us to explore and understand the data on Covid-19 in Indonesia in a more intuitive and visually appealing way (Schneiderman, 1996; Seaborn, 2021).

In this report, we use Tableau and Python to produce understandable and accessible visualisations of Covid-19 data in Indonesia. We'll concentrate on historical patterns, the geographic distribution of cases, and the effects of different interventions. We aim to support more informed decision-making and assist in reducing the spread of the virus in Indonesia by offering simple and understandable visuals.

**PROBLEM STATEMENT**

Covid-19 has had a significant impact on Indonesia, with the virus spreading rapidly throughout the country. It is important to understand the trajectory of the pandemic in Indonesia in order to inform decision-making and implement effective strategies to mitigate its spread.

One way to visualize and understand the data on Covid-19 in Indonesia is through the use of Tableau. However, the available data on Covid-19 in Indonesia is often complex and may be presented in a variety of formats, making it difficult to accurately and efficiently analyse and understand.

The problem we aim to address is the lack of clear and easily accessible visualizations of Covid-19 data in Indonesia, specifically using Tableau. We aim to create interactive dashboards in Tableau that allow users to easily explore and understand the data on Covid-19 in Indonesia, including trends over time, geographical distribution of cases, and the impact of various interventions. By providing clear and intuitive visualizations, we hope to facilitate more informed decision-making and help mitigate the spread of the virus in Indonesia.

**LITERATURE REVIEW**

Data visualization has become an increasingly important tool for understanding and analysing complex data sets, particularly in the context of the Covid-19 pandemic, Visualization tools such as Tableau and Python allow users to create interactive dashboards and graphs that enable them to explore and understand data in a more intuitive and visually appealing way (Shneiderman, 1996; Seaborn, 2021).

One study (Wang et al., 2020) used spatial mapping and dashboard visualization to track the spread of COVID-19 in Indonesia, specifically in Jakarta. The authors found that the visualization tools were effective in identifying hotspots and trends in the spread of the virus, allowing for targeted interventions and policies.

However, there has been limited research on the use of Tableau and Python to visualize Covid-19 data in Indonesia specifically. This is particularly important given the unique characteristics and challenges of the Covid-19 pandemic in Indonesia, including a large and dispersed population, limited healthcare resources, and cultural and behavioural factors that may impact the spread of the virus (Ministry of Health, Republic of Indonesia, 2020).

In this review, we aim to summarize the existing literature on the use of Tableau and Python for visualizing Covid-19 data in Indonesia. By providing a comprehensive overview of the current state of the field, we hope to identify gaps in the literature and highlight potential directions for future research.

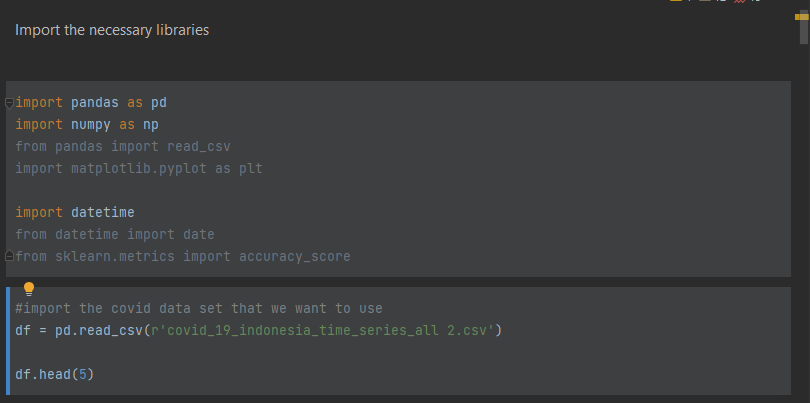
**METHODOLOGY**

To visualize the data on Covid-19 in Indonesia using Tableau and Python, we will follow the following methodology:

1. Data collection: We will collect data on Covid-19 in Indonesia from a variety of sources, including government websites, international organizations, and published research articles. The data will include information on confirmed cases, deaths, recoveries, total deaths, and other relevant variables.
2. Data cleaning and pre-processing: We will clean and pre-process the data to ensure that it is in a usable format for analysis. This will involve tasks such as removing missing values, standardizing variables, and training datasets.
3. Data exploration and visualization: We will use Tableau and Python to explore and visualize the data on Covid-19 in Indonesia. This will involve creating interactive dashboards and graphs that allow users to explore the data in a variety of ways, including by time period, province, and intervention.
4. Training and testing: We will validate and test the visualizations to ensure that they are accurate and effective in communicating the data. This will involve reviewing the visualizations for errors and inconsistencies.

**DATA COLLECTION**

The covid 19 dataset used for this project was gotten from Kaggle website.

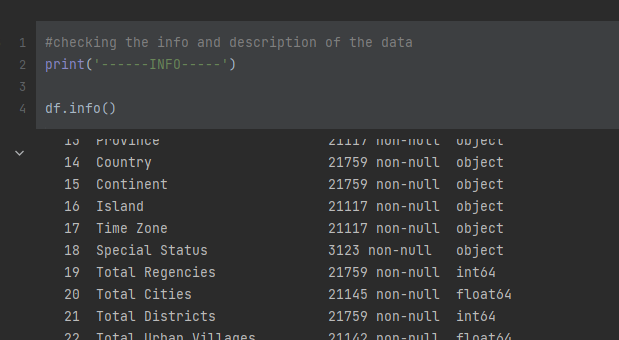


So the necessary libraries was imported, Using the read csv function from the pandas library, the code just created reads data from a CSV file and produces a Pandas data frame from it. The first five rows of the data frame will be shown by the df.head(5) command.

Graphical user interface, text

Description automatically generated

**Data cleaning and pre-processing**

Before exploring the data, had to make sure the dataset has no missing values by using the ISNULL () function and then get the info and the shape of the dataset using the code below. The data frame's data types, the number of non-null values in each column, and its memory usage are all listed in the df.info() method's summary of the data frame. 

Text

Description automatically generated

Several operations are carried out on the data frame by the code written above. A few columns are first removed from the data frame using the drop method. In contrast to the row axis (axis=0), the column axis (axis=1) is the direction in which the columns should be removed. This direction is specified by the axis parameter.

Next, the code defines a list of new column names and assigns it to the **header\_names** variable. These column names will be used to replace the existing column names in the data frame.

The Simple Imputer class is then imported from the sklearn.impute module and a new instance of the Simple Imputer class is created. Using the most frequent value for that column as a replacement, this class can be used to impute missing values in a dataset.

The code then imports the seaborn and matplotlib.pyplot libraries before using Seaborn’s boxplot function to plot the data in the data frame in a box. The data argument indicates the data that should be used to construct the figure, and the figsize parameter specifies the size of the plot.

Finally, the plot is displayed by the code using the show function from matplotlib.pyplot.

The box plot is a visual representation of the data's distribution that can be used to spot outliers and evaluate the data's basic shape.

Graphical user interface, application

Description automatically generated

The population column is the only column that has outliers in the data frame.

Text

Description automatically generated

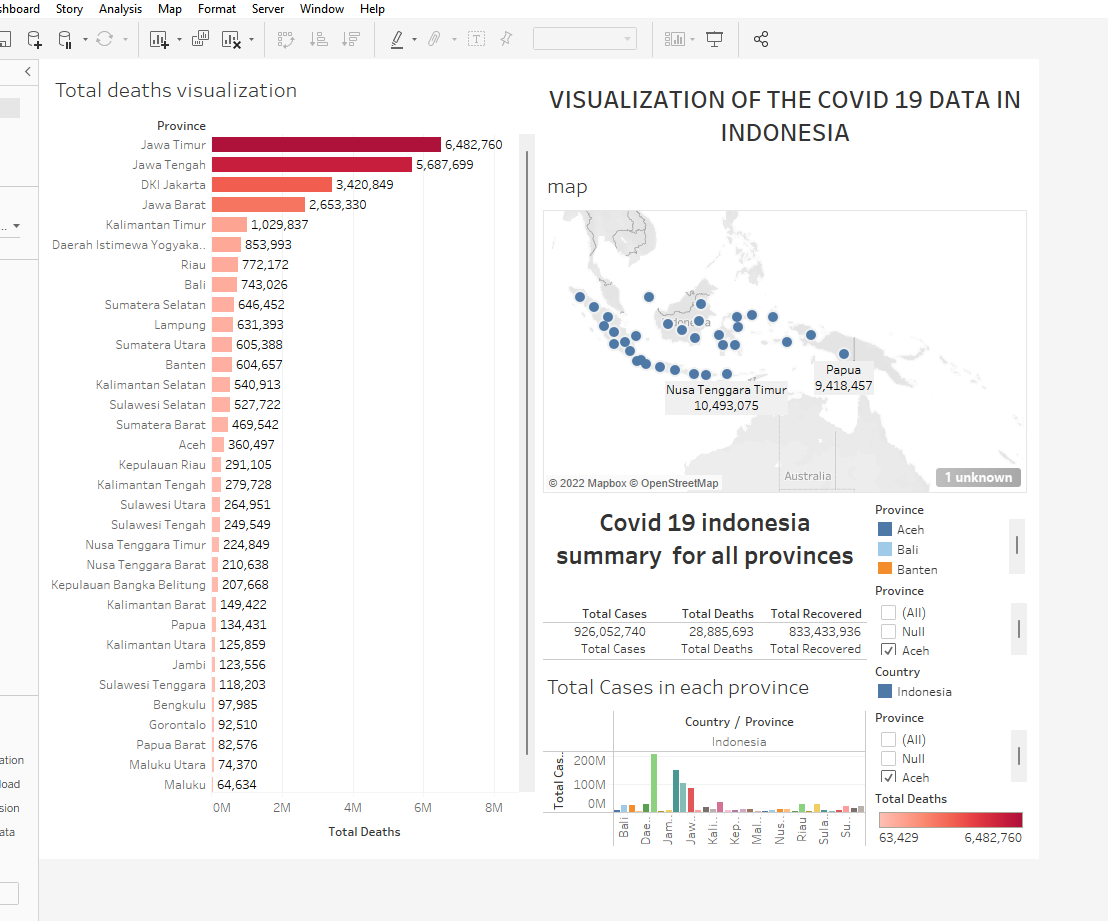
The data frame's rows with null values are removed by the code above. This is accomplished by identifying rows with null values using the isnull method and removing such rows from the data frame using the drop method, The null value identification axis is specified by the axis parameter. Axis=1 in this situation directs that null values be noted along the column axis rather than the row axis.

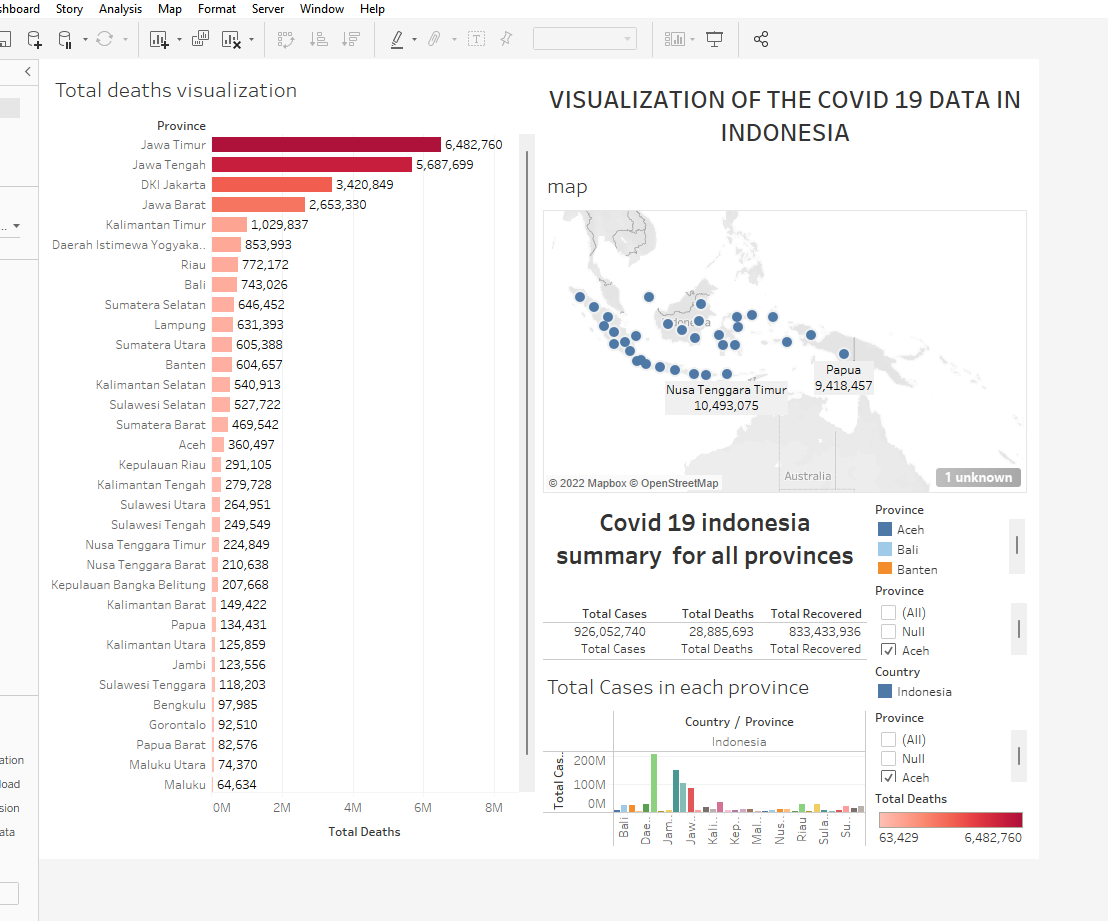
The missing values in the data frame are then imputed using an instance of the Simple Imputer class that is created later in the code. In this situation, the most frequent value in the column will be used to fill in for any missing values, according to the most frequent imputation strategy specified by the strategy argument.

The imputer is then fitted to the data using the fit method, and the imputation is applied to the data using the transform method. The pd is then used to transform the output data back into a data frame. The column names for the new data frame are specified using the header names list and the Data Frame function.

**Data exploration and visualization**

Tableau was used to create an interactive dashboard which shows the overview representation of the dataset, Had to make sure that the dataset has these metrics in order to generate visualisations that show the total values for each metric through time and a dashboard that shows the total cases, total deaths, and total recoveries for COVID-19 in Indonesia. A side bar graph might be used to display the total cases in each province, while a bar graph could display the total recoveries.

****

**Dashboard using tableau**

Additionally, analyze the scatter plot's scatter plot's correlation between the variables from the data distribution using python.

The pairplot function from the Seaborn library is used in the code below to generate a scatter plot matrix for the data2 data frame. A grid of plots called a scatter plot matrix is used to display the correlation between numerous pairs of variables.

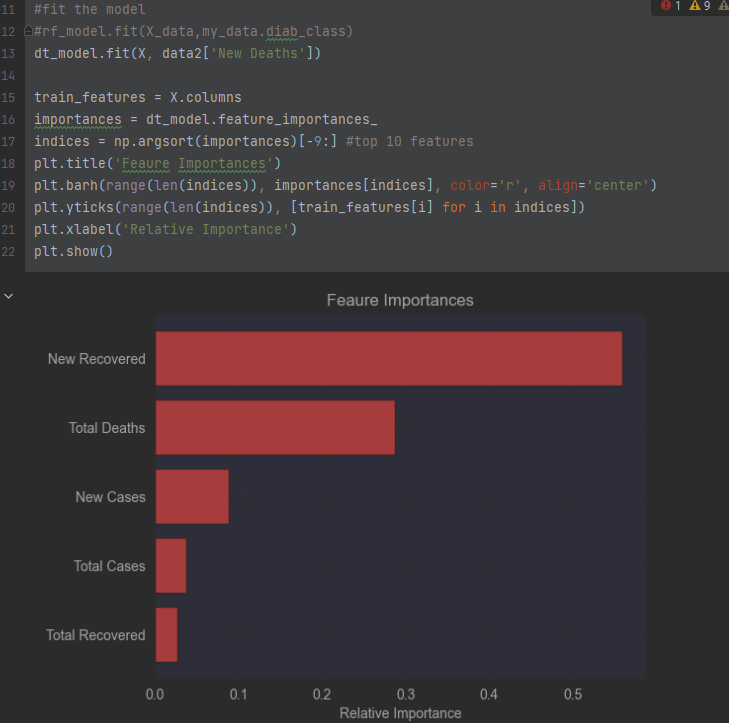
For each pair of variables in the data frame, the pairplot function generates a scatter plot, which is then shown on a grid. In order to display the distribution of each variable along the diagonal of the grid, histograms or kernel density plots are also used.

A picture containing text, blackboard, display

Description automatically generated

**Feature Importance**

Had to check which variable or feature contributes more to the prediction of the New Deaths targets variable.



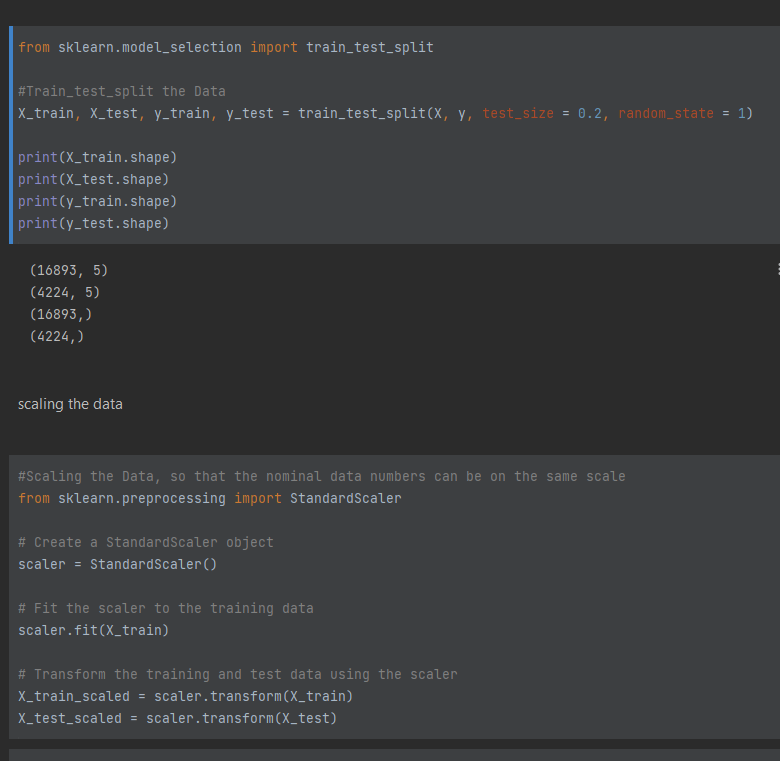
This code applies a decision tree model to the input data X and the target data data2['New Deaths'] using the DecisionTreeRegressor or DecisionTreeClassifier from scikit-learn. The feature importances\_ attribute of the model is used to obtain feature importances, which are then calculated using the model.

The top 10 features, as determined by the importance, are then displayed on the bar chart, and the feature importance are then plotted using matplotlib.

From the result above, we could tell that the New recovered variable, the total deaths and new cases are the most important features in the dataset in making predictions.

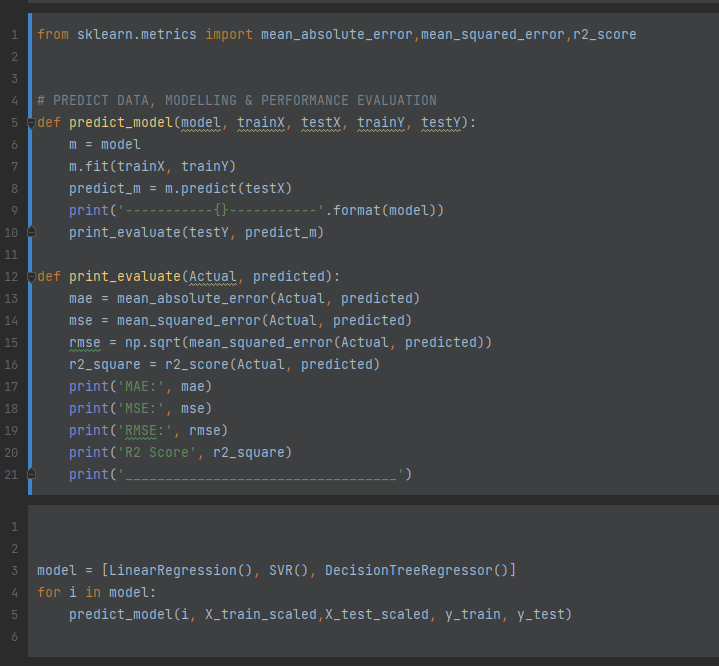
**MAKING PREDICTIONS AND EVALUATING THE MODEL PERFORMANCE**

Ensured the dataset is split into both train and test data on a ratio of 80:20 and then scaled the train data using the standard scaler function.



The StandardScaler object from the preprocessing module of Scikit-Learn is then used in the code to normalise the data. Data must be transformed during standardisation such that their mean and variance are both zero. This can be helpful in some machine learning methods since it can stop particular features from overpowering the model because of their size.

The StandardScaler object is initially fitted to the practise data using the fit method in order to normalise the data. The transformed training data is then saved in X train scaled after the training data has been modified using the transform technique. The scaler is also used to alter the test data, and the modified test data is then saved in X test scaled.



The program's predict model function uses the model's predict method to fit each model to the training data and generate predictions about the test data. The print evaluate function uses the necessary scikit-learn algorithms to calculate the MAE, MSE, RMSE (the square root of MSE), and R2 score after receiving the predictions. The console is then printed with the results.

The performance of many models on a given dataset may be compared using this code, and the model that performs the best based on these evaluation measures can be chosen. It's critical to remember that the specific problem being solved and the environment in which the model is being used can influence the choice of assessment measures.

**CONCLUSION**

Using Tableau and Python, it is possible to visualise and anticipate the spread of COVID-19 throughout Indonesia and learn more about its potential effects. With the use of these tools, it is possible to examine and spot patterns and trends in the data and forecast how things will pan out in the future.

Overall, using Tableau and Python can be a helpful tool for comprehending the transmission and effects of COVID-19 in Indonesia as well as for making defensible choices regarding how to handle the epidemic. It is crucial to remember, however, that every study or forecast should have its findings thoroughly evaluated and understood in light of the particular issue being addressed and the data being employed.

References:

* Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Belmont, CA: Wadsworth International Group.
* Ministry of Health, Republic of Indonesia. (2020). Covid-19 in Indonesia. Retrieved from <http://www.kemkes.go.id/article/view/20010400001/covid-19-di-indonesia.html>
* Seaborn. (2021). Seaborn: Statistical data visualization. Retrieved from <https://seaborn.pydata.org/>
* Schneiderman, B. (1996). The eyes have it: A task by data type taxonomy for information visualizations. In Proceedings of the 1996 IEEE Symposium on Visual Languages (pp. 336-343). Los Alamitos, CA: IEEE Computer Society Press.
* TENDA, E.D. et al., 2021. The COVID-19 surge in Indonesia: what we learned and what to expect. Breathe, 17(4), 210146
* Wang, D., Purnomo, S. Y., & Suryanto, E. (2020). Spatial Mapping and Dashboard Visualization of COVID-19 in Jakarta, Indonesia. Sains Humanika, 2(2), 21-30.
* World Health Organization. (2020). Covid-19 in Indonesia. Retrieved from <https://www.who.int/countries/idn/en/>