Capstone project

Software Technology 1

U3261301

Muhammad Moaz

Table of Contents

[Introduction 1](#_Toc133708957)

[Methodology 1](#_Toc133708958)

[Stage 1: Algorithm Design Stage 1](#_Toc133708959)

[Dataset Description 1](#_Toc133708960)

[Exploratory Data Analysis 1](#_Toc133708961)

[Predictive Data Analysis 1](#_Toc133708962)

[Model Preperation and Development 2](#_Toc133708963)

[Stage 2: Algorithm Implementation Stage 2](#_Toc133708964)

[Stage 3: Software Deployment Stage 2](#_Toc133708965)

[Conclusions 2](#_Toc133708966)

[References 2](#_Toc133708967)

# Introduction

This report describes the details of my Python Capstone Project for the Software Technology 1 unit within the scope of the project requirements provided on the assignment page. I have decided to work on the project using a country gross domestic product dataset available in Kaggle repositories.

A countries gross domestic product (GDP) is the total market value of all goods and products sold within a country. It is most often used by countries to measure their economic health and to compare the economies of countries. A countries GDP per capita which is a sub section of the countries GDP, is the gross value contributed to a country's GDP by each resident producer in the economy. A country’s GDP is a core indicator of a country’s economic performance and is reflective of a country’s living in standards, specifically in terms of cost of living.

Currently, predicting the continent on which a country resides is unreliable due to the large economic range of countries on a continent. For example, each continent (excl. Antarctica ) has an average of 33 countries of which there are both low GDP and high GDP countries. This makes it hard to correctly predict the continent, therefore, it would be beneficial to develop a software tool that utilises columns in the country GDP dataset to correctly predict the continent.

This report presents the details of the development of the software tool, exploring sections such as the exploratory data analysis (EDA), predictive analytics (PDA), implementation and deployment for this capstone project. Further, to assist in streamlining the EDA process, I proposed the following five questions:

1. What is the distribution of GDP per capita across different continents?
2. Is there a relationship between a country's population and its GDP per capita?
3. Are there any notable differences in the GDP estimates reported by the IMF and UN for different countries?
4. What is the overall distribution of population across different continents?
5. Is there a correlation between a country's rank (in terms of population, GDP, or other measures) and its continent?

The details of the methodology used is presented in the next section.

# Methodology

The methodology I used to develop the software tool consists of the steps outline below:

1. Development of models on EDA and PDA to identify the best performing models for solving the problem at hand. I plan to use the Pandas, matplotlib and scikit libraries. These libraries provide a vast range of methods to help visualise the dataset and provide metrics to measure the accuracy of the predictions.
2. Implementation of the best performing model as a desktop Tkinter software tool.
3. Deployment of the tool as a web platform tool using Flask.

## Stage 1: Algorithm Design Stage

The algorithm design stage consists of 3 main sections, these are the EDA, the PDA and the model preparation and development. Each section is explored in the following subsections.

### Dataset Description

The selected dataset called ‘Country\_GDP’ provides a comprehensive collection of data on 212 countries, including their ID, population, GDP estimates from the International Monetary Fund (IMF) and United Nations (UN), GDP per Capita, Global GDP rank, and continent. The dataset was sourced from Kaggle and can be found at the following link: <https://www.kaggle.com/datasets/ppb00x/country-gdp>. The dataset consists of 212 rows and 8 rows in which the data is both numerical and non-numerical depending on the column.

### Exploratory Data Analysis

Before performing an EDA on my dataset, I performed a short planning stage in which I decided to split the EDA into precise EDA two sections, the general EDA, and the. The aim of the general EDA was to gain a good initial understanding of the dataset, while the aim of precise EDA was to gain a more detailed understanding of the dataset and to answer the five questions I mentioned in the introduction. Before developing the visualisations for the EDAs, I started by understanding the basic description of the data and preparing the dataset, the method for this can be seen below.

A screenshot of a computer program

Description automatically generated with medium confidence

The output for the above code can be seen below. The output consists of the first 5 rows of the dataset, the las 5 rows of the dataset, the shape of the dataset (rows and columns) and the dataset’s statistics which is things like the mean, standard deviation, count, minimum, maximum, and more.

A screenshot of a computer

Description automatically generated

#### General EDA

The general EDA consisted of three graphs in which I identified the trends and patterns to gain a basic understanding of the database. The method and results for the three visualisations are as follows.

The first visualisation was a visualisation of the distribution of GDP per capita. I visualised this by using the matplotlib library to create histogram where x axis was the GDP per capita, and the y axis was the count. The histogram can be seen below.

A picture containing text, screenshot, diagram, rectangle

Description automatically generated

A trend identified in this visualisation was that the distribution of GDP per capita is right skewed, this indicates that there are few countries with very high GDP per capita values, but a larger number of countries with lower GDP per capita values.

The second visualisation was a comparison of population, and the IMF GDP estimates to find any relationships between them. I visualised this by using the matplotlib library to create a scatter plot where the x axis is population, and the y axis is the IMF GDP estimates. The scatter plot can be seen below.

A picture containing text, screenshot, diagram, plot

Description automatically generated

A trend I identified from this scatter plot is that there is a positive relationship between a country’s population and its GDP according to the IMF estimates. As the population increases, the GDP tends to increase as well. However, there are some outliers with high population values and relatively low GDP values.

The third visualisation was a correlation matrix heatmap of rank, population, both GDP estimates (UN and IMF) and GDP per capita. To do this I created a list with the specified columns and used it with the matplotlib library to create a correlation matrix heatmap. You can see the heatmap below.

A picture containing text, screenshot, square, colorfulness

Description automatically generated

I identified several trends and patterns through this heatmap, the trends are as follows:

* There is a strong positive correlation between population and both IMF and UN GDP estimates, indicating that countries with larger populations tend to have a higher GDP value.
* There is also a strong positive correlation between IMF and UN GDP estimates, suggesting that the two organizations' estimates are similar.
* There is a moderate positive correlation between population and GDP per capita, indicating that countries with larger populations tend to have higher GDP per capita.
* There is a weak negative correlation between rank (in terms of population, GDP, or other measures) and continent, indicating that there is no clear relationship between a country's rank and the continent it is located in.

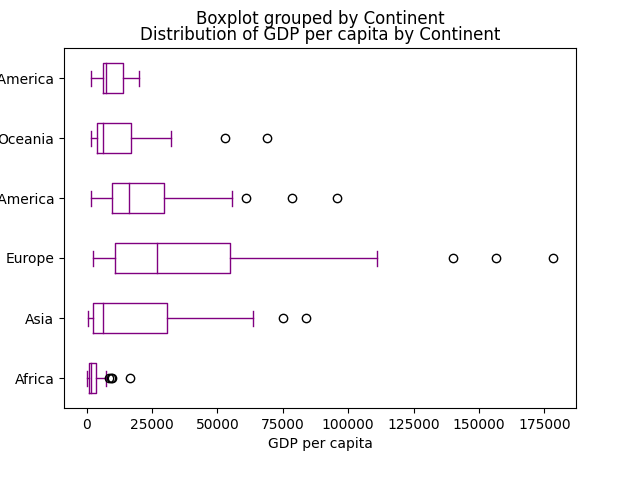
#### Precise EDA

In the precise EDA I created visualisations in the form of graphs to answer each of the five questions. I used a wide variety of graphs, such as a scatter plot, box plot and bar chart to visualise the data.

To answer the first question, ‘What is the distribution of GDP per capita across different continent?’, I created a boxplot using the matplotlib library where the x axis was the GDP per capita, and the y axis was the continent. The method and result of this visualisation can be seen in the below figures.

A screen shot of a computer code

Description automatically generated with low confidence



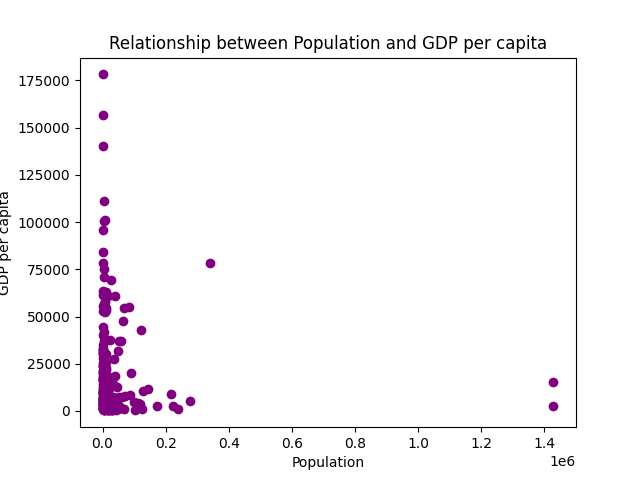
I identified three trends by analysing this box plot, the trends are as follows:

* The distribution of GDP per capita varies greatly across different continents.
* North America and Europe have the highest GDP per capita on average, while Africa has the lowest.
* The range of GDP per capita is widest in Asia, indicating greater variability in the GDP per capita of countries within the continent.

To answer the second question, ‘Is there a relationship between a country's population and its GDP per capita?’, I created a scatter plot graph where population was on the x axis and GDP per capita was on the y axis. The method and results of the scatter plot can be seen below:

A screen shot of a computer code

Description automatically generated with low confidence



The trends I identified in this scatter plot are:

* There is a positive correlation between a country's population and its GDP per capita.
* However, this relationship is not necessarily linear - the highest GDP per capita values are generally associated with countries with populations between 0 and 1 billion people.
* There are some outliers where countries with relatively low populations have very high GDP per capita values (such as Qatar).

Next, to produce a result for the third question, ‘Are there any notable differences in the GDP estimates reported by the IMF and UN for different countries?’. I used matplotlib to create another scatter plot this time with the IMF GDP estimate on the x axis and the UN GDP estimate on the y axis. This plot would explore the relationship between the two GDP estimates. The method and results of the scatter plot can be seen below.

A screen shot of a computer code

Description automatically generated with low confidence

A picture containing text, screenshot, diagram, plot

Description automatically generated

The trends and patterns I identified for this scatter plot are that:

* There is generally good agreement between the IMF and UN GDP estimates, with most points falling near the diagonal line.
* However, there are some notable differences, particularly for countries with very high or very low GDP estimates.
* The colorbar shows that the differences between the IMF and UN GDP estimates tend to be small, with only a few outliers with larger differences.

To answer the fourth question, ‘What is the overall distribution of population across different continents?’, I created a bar chart of population by continent using the matplotlib library. This meant that the continent was on the x axis and population was on the y axis. The resulting bar chart and its Python code can be seen below.

A screen shot of a computer code

Description automatically generated with low confidence

A picture containing text, diagram, screenshot, plot

Description automatically generated

The following are the trends I identified in the visualisation for question 4:

* Asia has the highest total population by a significant margin, followed by Africa.
* North America and Oceania have the lowest total population.
* The total population for each continent is roughly proportional to the number of countries represented in the dataset for that continent.

The final visualisation for the precise EDA was for question 5, ‘Is there a correlation between a country's rank (in terms of population, GDP, or other measures) and its continent?’. To produce an answer for this question, I decided to create a scatter plot using the matplotlib library which compares the x axis, rank and the y axis, continent. The visualisation in the form of a scatter plot and the code can be seen below.

A screen shot of a computer program

Description automatically generated with low confidence

A picture containing text, screenshot, line, font

Description automatically generated

The trends and patters identified in the scatter plot are that:

* There is no clear relationship between a country's rank and its continent - countries from all continents are represented at every rank.
* However, there are some interesting outliers, such as China and India, which have much higher population ranks than any other country in their respective continents.
* Europe has a higher concentration of high-ranked countries compared to other continents.

### Predictive Data Analytics

Once I gathered a detailed understanding of the dataset, it’s trends and patterns, I started on the predictive data analytics (PDA) stage. Before developing the PDA, I had to propose an aim and so the aim I proposed was to create a model which predicts the continent of a country given the population, GDP per capita, IMF GDP estimate and UN GDP estimate. The PDA consists of a couple of sections, these are pre-processing, classifier comparison to identify the best machine learning model and performance evaluation with a variety of metrics.

Pre-processing is the act of processing data to make it suitable for the predictive analytics stage. In my case, the dataset contains both numerical and non-numerical data, and so to make it suitable for the PDA, I encoded the non-numerical values using the label encoding method. This can be seen below.

A screen shot of a computer code

Description automatically generated with low confidence

### Model Preperation and Development

The model preparation and development stage consisted of many steps. I started off splitting my data into features and targets, in my case, the features were the population, IMF GDP, UN GDP and GDP per capita, so I declared them by dropping the columns I didn’t need. On the other hand, the target for my model was the continent and so I declared it as such. After this, I split my data into training and testing sets, I did this by using the ‘train\_test\_split’ function and inputting the target and features as parameters. I also set the test size parameter to 0.2, this means that a random 80% of the data would be used for training the model while the remaining 20% would be used for testing the model.

A screen shot of a computer

Description automatically generated with medium confidence

Next, I initialise the models I would be training to predict the continent, for this section I used the following models:

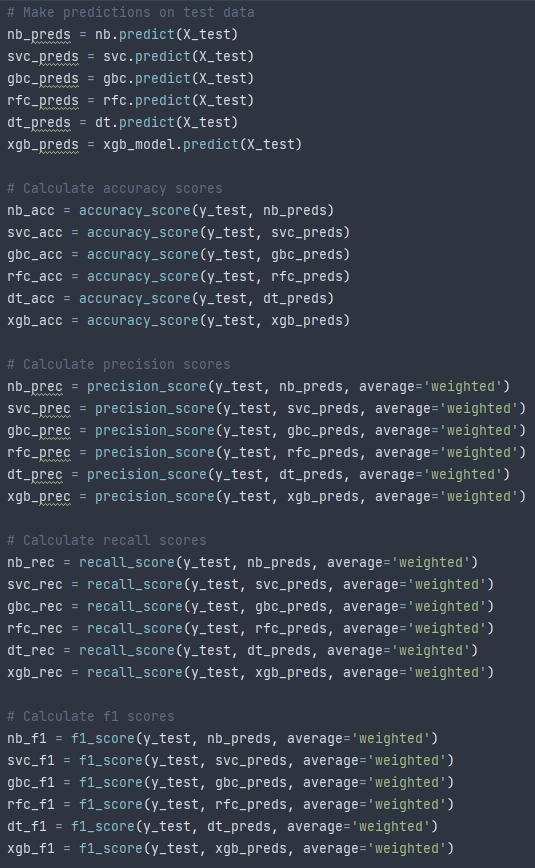
* Gaussian Naïve Bayes
* Support Vector
* Gradient Boosting
* Random Forest
* Decision Tree
* XGBoost

Once I initialised the models, I fit each model to the training data. This meant that each model would be trained on the dataset using a random 80% of the data. This can be seen below:

A screenshot of a computer program

Description automatically generated with medium confidence

After initialising and training the models, it was time for the models to make a prediction on the test data and then measure the predictions using metrics such as accuracy scores, precision scores, recall scores and f1 scores.



Once I had collected results for each model’s prediction in each metric, it was time too look at the results and decide which model had performed the best. Looking at the following results, we can see that the XGBoost model produced the best accuracy score with a score of 37.21%.

A screenshot of a computer

Description automatically generated

Now, a score of 37.21% isn’t the greatest and is considered on the lower side of the range, however it is the best performing model and so we can use the pickle library to save the model as a file called ‘best\_model.pkl’. Saving the model is important as we can then load it and use it in both the software and web application implementation.

A screen shot of a computer program

Description automatically generated with low confidence

The final stage in the PDA was the model performance evaluation stage, it is similar to the metrics in we measure for each model in the above stage, however, it is focused on the best performing model to gain an understanding of the model. The first method I used to evaluate the model’s performance was by creating a confusion matrix. A confusion matrix is a table which defines the performance of a classification algorithm, which in this case is the XGBoost model.

A screen shot of a computer code

Description automatically generated with low confidence

A screenshot of a computer screen

Description automatically generated with medium confidence

The second method to evaluate the performance was a classification report, I implemented the classification report with the following code, and the second image was its output.

A screen shot of a computer

Description automatically generated with low confidence

A screenshot of a computer

Description automatically generated with medium confidence

I also tried to implement a prediction report, however there was an error in the code, that I failed to debug in time and so I wasn’t able to include it in the report.

In the future, if I ever get another opportunity, I will spend more time trying to increase the accuracy score of the model as I know there are techniques which can be performed to do so. However, due to not being experienced enough and bad time management, I wasn’t able to try these techniques on my prediction model.

## Stage 2: Algorithm Implementation Stage

The algorithm implementation stage consists of two aspects:

1. The implementation of the best performing machine learning model as a desktop Tkinter software tool.
2. Deployment of the tool as a web enabled platform tool.

This section explores the first, it explores how I implemented the best performing prediction model into a desktop Tkinter software tool.

The best performing machine learning model from the previous section was the XGBoost model. I started the Tkinter GUI implementation of the prediction model by first loading the model we saved using pickle.

A picture containing text, font, screenshot

Description automatically generated

Once the model was loaded, I started working on the GUI. My GUI consisted of four label and entry widgets which asked the user for the population, IMF GDP, UN GDP, and GDP per capita inputs. I also packed the widgets using grid, so that I can organise them on the window. I didn’t use any frames as the GUI was simple and using frames would’ve added complexity to it.



After creating the widgets, I created the predict and quit buttons. The predict would call the predict function which uses the trained model to predict the continent given the user’s input, uses the continent dictionary to find the continent value from the prediction value and outputs the continent, while the quit button destroys the tkinter environment and exits the program. The code for the buttons and the predicts function can be seen below.

A screen shot of a computer program

Description automatically generated with low confidence

A screen shot of a computer program

Description automatically generated with low confidence

Putting all this code together, produces the desktop software implementation of the best performing model in the form of a Tkinter GUI program. The resulting GUI can be seen below.

A screenshot of a computer

Description automatically generated

The GUI in action can be seen in the following example, in this example, I enter the inputs relating to Qatar, a country in Asia. When I click the predict button, we can see that the model correctly predicts the continent and outputs it as well.

A screenshot of a computer

Description automatically generated

## Stage 3: Software Deployment Stage

The deployment of the software as a desktop tool limits its applicability and does not allow for a wider usage. Hence there is a need to deploy this software as a web-based tool. I performed the deployment of the continent prediction tool as a web-based platform using Flask API, a widely used micro web framework for creating APIs in Python.The Flask project deployment for the continent prediction tool is available at the link below.

I implemented the best performing model as a web application by first importing all the required libraries.

A picture containing text, font, screenshot

Description automatically generated

Then I initialised a Flask application and loaded the best performing model using the Pickle method load.

A screen shot of a computer

Description automatically generated with medium confidence

After that I defined the home page route which renders the html file, I created in the template’s directory.

A screenshot of a computer

Description automatically generated with medium confidence

Next, I defined the prediction page route where I declared the predict function. The predict function declares the continent dictionary, where the keys are a value from 0 to 5 and the values are the continent. Also in the prediction page route, I created a data frame with the input features, used the trained model to make a prediction with the data frame, and then returned the predicted continent to the html file containing the web app.

A screen shot of a computer program

Description automatically generated with low confidence

After creating the prediction page route, I wrote some code which would run the application.

A picture containing text, font, screenshot, design

Description automatically generated

The code for the HTML page including an internal style sheet can be seen below. I decided to implement a style sheet to the HTML as without, the continent prediction tool looks very rushed and unprofessional, which is not the aesthetic I am aiming for.



Running the web application Python file starts a server on which the web application runs. The web application can be seen in the below image.

A screenshot of a computer

Description automatically generated with medium confidence

To test the deployed web application, we can enter in the details of a country and see if it correctly predicts the continent on which the country lies. For example, we can use the details of New Zealand, a country in Oceania.

A screenshot of a computer

Description automatically generated with medium confidence

As we can see, the web application correctly predicts the continent. Therefore, the test was successful, and the web application is operating as intended.

# Conclusions

This report covers the design, development, implementation, and deployment process of a data driven continent prediction software platform using Python for the ST1 Capstone Project. As can be seen from the outcomes of this project, we can train models that can predict the continent with substantial agreement with the population, GDP per capita, IMF GDP and UN GDP. The availability of predictive analytics tools as both desktop software tool and a web-based tool allows for wider application of this project.

# References

OECD (2022). *GDP and spending - Gross domestic product (GDP) - OECD Data*. [online] OECD. Available at: https://data.oecd.org/gdp/gross-domestic-product-gdp.htm.

P., S. (2023). *Country\_GDP*. [online] www.kaggle.com. Available at: https://www.kaggle.com/datasets/ppb00x/country-gdp [Accessed 12 Apr. 2023].

The Investopedia Team (2022). *What Is Per Capita GDP?* [online] Investopedia. Available at: https://www.investopedia.com/terms/p/per-capita-gdp.asp#:~:text=GDP%20per%20capita%20measures%20the [Accessed 4 May 2023].