|  |  |
| --- | --- |
| **Programme** | BSc (Hons) Data Scientist |
| **Module name** | Data Science Professional Practice |
| **Schedule Term** | January 2024 Term – Sep 2023 Group 1 Cohort |
| **Student Reference Number (SRN)** | BP0289018 |
| **Report/Assignment Title** | Data Science Professional Practice – Public Data Exercise |
| **Date of Submission**  ***(Please attach the confirmation of any extension received)*** | dd/mm/2024 |
|  | |
|  | |
| Declaration of Original Work:  I hereby declare that I have read and understood BPP’s regulations on plagiarism and that this is my original work, researched, undertaken, completed and submitted in accordance with the requirements of BPP School of Technology.  The word count, excluding contents table, bibliography and appendices, is xxxx words.  Student Reference Number: BP0289018 Date: dd/mm/2024 | |
| By submitting this coursework you agree to all rules and regulations of BPP regarding assessments and awards for programmes. **Please note, submission is your declaration you are fit to sit.**  BPP University reserves the right to use all submitted work for educational purposes and may request that work be published for a wider audience.  **BPP School of Technology** | |

**Level 6 (BSC Hons) Data Science Apprenticeship**

**Data Science Professional Practice**

**Public Data Exercise**

Contents

[Table of Figures 3](#_Toc169197360)

[Formative – Notes/Guidance for assessor 4](#_Toc169197361)

[Introduction 5](#_Toc169197362)

[Tools 6](#_Toc169197363)

[Data 6](#_Toc169197364)

[Data Features 7](#_Toc169197365)

[ETL Pipeline 7](#_Toc169197366)

[Data Quality and Exploratory Data Analysis (EDA) 7](#_Toc169197367)

[Data Analytics 14](#_Toc169197368)

[Linear Regression Model 14](#_Toc169197369)

[Moving Average (ARIMA) Model 16](#_Toc169197370)

[Reflections and Next Steps 18](#_Toc169197371)

[References 19](#_Toc169197372)

[Bibliography 19](#_Toc169197373)

# Table of Figures

[Figure 1: Simple ETL diagram 7](#_Toc169197331)

[Figure 2: Data Quality Principles 8](#_Toc169197332)

[Figure 3: Import necessary Python libraries 8](#_Toc169197333)

[Figure 4: Import source data 8](#_Toc169197334)

[Figure 5: Code to preview the source data 9](#_Toc169197335)

[Figure 6: Output of the initial preview of source data 9](#_Toc169197336)

[Figure 7: code for filtering on date and updating date format 9](#_Toc169197337)

[Figure 8: Second preview of source data 10](#_Toc169197338)

[Figure 9: Code for a summary of the source data 10](#_Toc169197339)

[Figure 10: Description of data types and counts 10](#_Toc169197340)

[Figure 11: Summary of data for each field 11](#_Toc169197341)

[Figure 12: Code to confirm no nulls in source data 11](#_Toc169197342)

[Figure 13: Output of nulls check 11](#_Toc169197343)

[Figure 14: Code to plot average temperature over time 12](#_Toc169197344)

[Figure 15: Plots of average temperature over time 12](#_Toc169197345)

[Figure 16: Code to plot a distribution chart for average temperatures 12](#_Toc169197346)

[Figure 17: Distribution charts of average temperature 12](#_Toc169197347)

[Figure 18: Code for generating a heatmap of average temperature 13](#_Toc169197348)

[Figure 19: Heatmaps of average temperature over years 13](#_Toc169197349)

[Figure 20: Code for linear regression model 14](#_Toc169197350)

[Figure 21: Output of evaluation of linear regression model 14](#_Toc169197351)

[Figure 22: Code to plot our Linear Regression model 15](#_Toc169197352)

[Figure 23: Output of Linear Regression model 15](#_Toc169197353)

[Figure 24: Code to create a Time Series Analysis model 16](#_Toc169197354)

[Figure 25: Evaluation of the model 16](#_Toc169197355)

[Figure 26: Code to plot ARIMA predictions against actual data 17](#_Toc169197356)

[Figure 27: Plot of ARIMA Time Series prediction 17](#_Toc169197357)

[Figure 28: Code to generate and display forward predictions 17](#_Toc169197358)

[Figure 29: Plots of forward predictions 17](#_Toc169197359)

# Formative – Notes/Guidance for assessor

Throughout this document, sections highlighted in yellow indicate where I intend to come back and review/refine, or indicate a direct ask for guidance from yourself.

My work-based impact-assessment will hopefully use my submission from term 2’s Data Visualisation assessment, and demonstrate data governance, ETL pipeline, and data visualisation. So for my exercise on public data I am intentionally leaning towards a data analytics challenge.

On reviewing ‘good’ examples, I can see Python is the favoured tool for these tasks. My coach has also advised me to get as much of a headstart as possible with Python given I have no experience using this in practice. So I have intentionally attempted to take the techniques shown to us in term 1 on excel, and carry them out in Python.

I have essentially looked on teaching websites for code to adapt, and therefore will list the key sites in my bibliography.

This is particularly true in the time series analysis. Our Data Analytics teacher took us through the principles of moving average prediction, showed us a few approaches in excel which I fully understand… but I have no idea what the Python functions are doing. I simply looked up a guide, followed the code, looked at the evaluations and output to validate it.

Can you advise if this approach looks acceptable?

# Introduction

The aim of this project was to apply Data Analysis techniques to existing global temperature data, in order to build a model that can be used to predict future temperature changes. Climate change is a massive concern across the world, presenting challenges for governments and populations. Being able to predict future changes in temperature would allow for data-led decision making and planning for things like infrastructure spending, future investments, research and much more.

I will attempt to apply two common data analytics approaches: Linear Regression based on an input factor, and Moving Average time series analysis based on past trends. After evaluating the merits of each approach, I will be able to make recommendations based on predicted trends.

## Tools

I will do this via Python, in a public Kaggle Notebook. Python is a language that is massively growing amongst the Data Science community as the tool of choice. It is incredibly versatile for data analytics and visualisation, machine learning, and has an extensive support and learning community.

*Insert quote supporting power of Python as a tool for references*

It is free to use for this sort of exercise via Kaggle, making it ideal as a learning and sharing exercise. The techniques I will apply are standard approaches that I have been taught via things like Excel and SQL, but part of this exercise is to build my own familiarity with such a powerful tool.

# Data

The data set I plan to use is the Earth Surface Temperature data from Kaggle.

Link to data: [Climate Change: Earth Surface Temperature Data (kaggle.com)](https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data)

## Data Features

Global Land and Ocean-and-Land Temperatures **(GlobalTemperatures.csv):**

* Date: starts in 1750 for average land temperature and 1850 for max and min land temperatures and global ocean and land temperatures
* LandAverageTemperature: global average land temperature in celsius
* LandAverageTemperatureUncertainty: the 95% confidence interval around the average
* LandMaxTemperature: global average maximum land temperature in celsius
* LandMaxTemperatureUncertainty: the 95% confidence interval around the maximum land temperature
* LandMinTemperature: global average minimum land temperature in celsius
* LandMinTemperatureUncertainty: the 95% confidence interval around the minimum land temperature
* LandAndOceanAverageTemperature: global average land and ocean temperature in celsius
* LandAndOceanAverageTemperatureUncertainty: the 95% confidence interval around the global average land and ocean temperature

## ETL Pipeline

The ETL process for this exercise is fairly straightforward, informed by the Data Quality checks and Exploratory Data Analysis EDA I will describe later

A diagram of a process

Description automatically generated

Figure 1: Simple ETL diagram

## Data Quality and Exploratory Data Analysis (EDA)

As we perform exploratory data analysis we should be mindful of the 6 principles of Data Quality as per the Government Data Quality Hub *Insert Reference* and look for opportunities to align to these:

Validity 
Completeness 
Data Quality 
Consistency 
Accuracy 
Uniqueness 
Timeliness 

Figure 2: Data Quality Principles

This helps ensure we can trust that any insight is based on trusted information, and ensure best performance of our models.

We start by importing necessary libraries:

A white background with black text

Description automatically generated

Figure 3: Import necessary Python libraries

Then import the data itself:

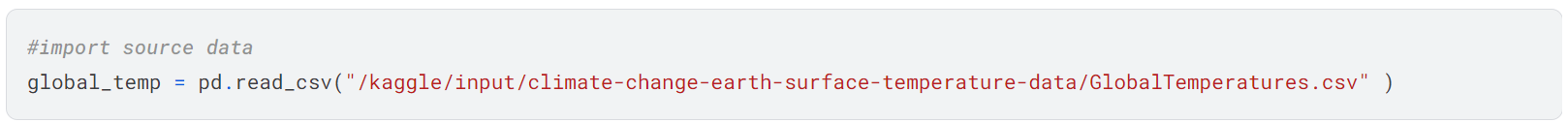


Figure 4: Import source data

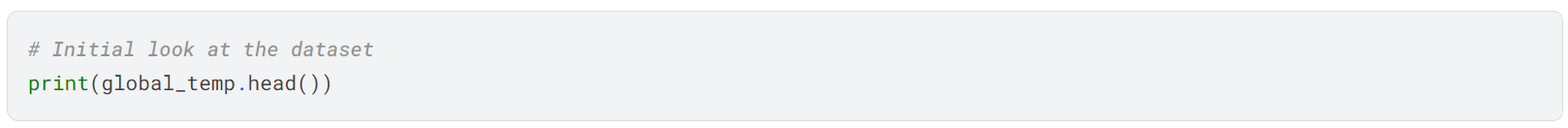
Now we can start to examine the data itself: 

Figure 5: Code to preview the source data

A screenshot of a computer

Description automatically generated

Figure 6: Output of the initial preview of source data

We can see immediately that the data appears to contain null values across the fields. This can be explained when we look at the description of the data itself, which states that data was only captured from 1850 for most fields.

This leads us to our first suggestion for preparing the data; to ensure that our data is complete and valid we should limit it to data after 1850. It also makes sense at this stage to reformat the date format to a more useable format, datetime:

A white background with black numbers

Description automatically generated

Figure 7: code for filtering on date and updating date format

Now when we preview the data, we can see all fields appear populated:

A screenshot of a computer

Description automatically generated

Figure 8: Second preview of source data

This all seems more within expectations, but we can do some further basic checks for data quality issues:

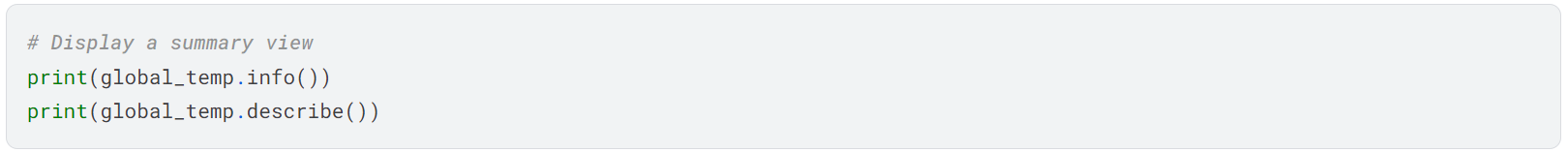


Figure 9: Code for a summary of the source data

A screenshot of a computer

Description automatically generated

Figure 10: Description of data types and counts

A screenshot of a computer

Description automatically generated

Figure 11: Summary of data for each field

These two checks let us see if the data is coming through in the format expected – which prevents calculation errors, and indicates any extreme or unusual values which could skew results.

There don’t appear to be any issues of either kind based on these views. We will do one final check to confirm we have removed nulls in the data:

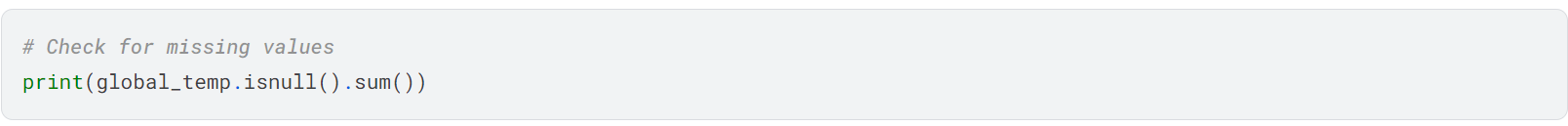


Figure 12: Code to confirm no nulls in source data

A screenshot of a computer

Description automatically generated

Figure 13: Output of nulls check

Our next EDA step is to visualise the data, which is a helpful way of spotting outliers, trends, common categories which may either prove useful in addressing the challenge or would point to some Data Quality issues that need addressed via our ETL process.

A close-up of a computer screen

Description automatically generated

Figure 14: Code to plot average temperature over time

A comparison of blue and white lines

Description automatically generated

Figure 15: Plots of average temperature over time

We can see an upwards trend in recent years, more pronounced when we include ocean temperatures. Also note the lower range is higher when ocean temperatures are included.

There is no indication of outliers or unusual activity in the data.

A close-up of a computer screen

Description automatically generated

Figure 16: Code to plot a distribution chart for average temperatures

A comparison of a graph

Description automatically generated

Figure 17: Distribution charts of average temperature

We see from these charts that the most common temperature is at the higher end, and again see that including ocean temperatures both increases the temperature and tightens the range. However, again there is no indication of outliers or unusual activity.

A white background with text

Description automatically generated

Figure 18: Code for generating a heatmap of average temperature

A screenshot of a graph

Description automatically generated

Figure 19: Heatmaps of average temperature over years

These heatmaps confirm our expectations: middle years are warmest, with recent years showing increasing warmth across the months. Again we see this trend more pronounced when we include ocean temperatures.

Our EDA did not highlight any outliers, unexpected trends or groupings. Therefore we can continue to modelling a prediction without any further data transformation.

# Data Analytics

## Linear Regression Model

Simple Linear Regression is one of the most common approaches to predicting an outcome based on one or more input variables. However to be meaningful, it depends on a strong correlation between the input variables and the output.

In this case we want to look at using the year as the input. Python allows us to easily build a model and train it based on a randomised 80/20 split of the data.

The final step in the code below will evaluate the model’s accuracy

A screenshot of a computer program

Description automatically generated

Figure 20: Code for linear regression model

Output:

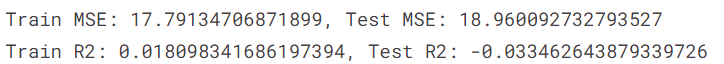


Figure 21: Output of evaluation of linear regression model

Unfortunately this output indicates that our model is not reliable. A high Mean Squared Error (MSE) tells us that the actual data points fall well away from our predictions, and an R-squared value away from 1 confirms that our predicted line largely misses the area around our data points.

This can be more easily confirmed visually.

A screen shot of a computer program

Description automatically generated

Figure 22: Code to plot our Linear Regression model

A graph showing the difference between the temperature and the temperature

Description automatically generated

Figure 23: Output of Linear Regression model

So a Linear Regression model would not appear to be a useful approach in solving our problem.

# Moving Average (ARIMA) Model

Autoregressive Moving Average is a popular approach to time series analysis in Python. It depends a series of data points with consistent properties, and makes predictions based on the change from one data point to the next over time, and modifying forward predictions based on the observed difference between actual data points and a moving average over a defined lag period.

Note to assessor: see note below about a simplified approach. Is this acceptable?

A screen shot of a computer code

Description automatically generated

Figure 24: Code to create a Time Series Analysis model

A screenshot of a computer

Description automatically generated

Figure 25: Evaluation of the model

The MSE here is fairly close to 0, indicating the model appears to be fairly reliable. We can confirm this more easily by plotting the predictions against the actuals.

A screen shot of a computer code

Description automatically generated

Figure 26: Code to plot ARIMA predictions against actual data

A screenshot of a graph

Description automatically generated

Figure 27: Plot of ARIMA Time Series prediction

We have produced a fairly reliable model for predicting future temperatures based on historic data, so lets project that forward:

A screen shot of a computer program

Description automatically generated

Figure 28: Code to generate and display forward predictions

A blue and red graph

Description automatically generated

Figure 29: Plots of forward predictions

This seems useful, though we must bear in mind ARIMA projections become increasingly limited as you look further ahead due to being fed by its own previous predictions.

# Reflections and Next Steps

We explored the data, performed some basic data transformation in line with Data Quality standards. EDA was carried out to identify patterns and outliers.

We tried two approaches to build a predictive model. Linear regression was not useful, but Moving Average Time Series allowed us to create reliable predictions.

I would recommend building this into a regular report, updated and refreshed via a governed automated pipeline given that moving averages can only be reliably extended so far.

As my source data misses recent years, it would be useful as a validation exercise to compare the model to actual results and in particular to understand how unexpected global events such as Covid lockdowns affected the trend.

We could revisit the usefulness of Linear Regression and other techniques by bringing other data around things such as location, population density and perhaps identify key contributing factors.

Given this is a public data set and platform we have not had to worry about data security and ethics, however if utilised in a decision-making process, with data down to local level, we may wish to consider things like masking the data.

# References

To be reviewed/updated

Anna Kott, 2023. *Enhancing Data Analytics for Heavy Industry Through UI/UK Design and Streamlined Reporting* [online] Available at: [Enhancing data analytics for heavy industry through UI/UX design and streamlined reporting (theseus.fi)](https://www.theseus.fi/bitstream/handle/10024/798445/Kot_Anna.pdf?sequence=5&isAllowed=y) [retreived April 2024]

Kaggle. (2017). *Climate Change: Earth Surface Temperature Data. [online]*. Retrieved June 2024, from Kaggle: https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data

Machine Learning Mastery. (2023). *How to Create and ARIMA Model for Time Series Forecasting. [online]*. Retrieved June 2024, from Machine Learning Mastery: https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

Statsmodels. (2024). *Time Series Analysis [online]*. Retrieved June 2024, from Statsmodels: https://www.statsmodels.org/stable/user-guide.html#time-series-analysis

Stirling, C. (2024). *DSPP Global Temperature Predictions*. Retrieved June 2024, from Github: https://github.com/CraigW-Stirling/DSPP-Global-Temperature-Predictions

UK Government. (2020). *The Government Data Quality Framework. [online]*. Retrieved June 2024, from https://www.gov.uk/government/publications/the-government-data-quality-framework/the-government-data-quality-framework

W3\_Schools. (2024). *Data Science & Python. [online]*. Retrieved June 2024, from W3 Schools: https://www.w3schools.com/datascience/ds\_python.asp

W3\_Schools. (2024). *Machine Learning - Linear Regression. [online]*. Retrieved June 2024, from W3 Schools: https://www.w3schools.com/python/python\_ml\_linear\_regression.asp

# Bibliography

To be reviewed/updated

Kaggle. (2017). *Climate Change: Earth Surface Temperature Data. [online]*. Retrieved June 2024, from Kaggle: https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data

Machine Learning Mastery. (2023). *How to Create and ARIMA Model for Time Series Forecasting. [online]*. Retrieved June 2024, from Machine Learning Mastery: https://machinelearningmastery.com/arima-for-time-series-forecasting-with-python/

Statsmodels. (2024). *Time Series Analysis [online]*. Retrieved June 2024, from Statsmodels: https://www.statsmodels.org/stable/user-guide.html#time-series-analysis

Stirling, C. (2024). *DSPP Global Temperature Predictions*. Retrieved June 2024, from Github: https://github.com/CraigW-Stirling/DSPP-Global-Temperature-Predictions

UK Government. (2020). *The Government Data Quality Framework. [online]*. Retrieved June 2024, from https://www.gov.uk/government/publications/the-government-data-quality-framework/the-government-data-quality-framework

W3\_Schools. (2024). *Data Science & Python. [online]*. Retrieved June 2024, from W3 Schools: https://www.w3schools.com/datascience/ds\_python.asp

W3\_Schools. (2024). *Machine Learning - Linear Regression. [online]*. Retrieved June 2024, from W3 Schools: https://www.w3schools.com/python/python\_ml\_linear\_regression.asp