**CCT College Dublin**

**Assessment Cover Page**

|  |  |
| --- | --- |
| Module Title: | *Programming for DA*  *Statistics for Data Analytics*  *Machine Learning for Data Analysis*  *Data Preparation & Visualisation* |
| Assessment Title: | *MSC\_DA\_CA2* |
| Lecturer Name: | *Sam Weiss*  *Marina Iantorno*  *Muhammad Iqbal*  *David McQuaid* |
| Student Full Name: | John O'Brien |
| Student Number: | SBS23011 |
| Assessment Due Date: | 26/05/23 |
| Date of Submission: |  |



**Declaration**

|  |
| --- |
| By submitting this assessment, I confirm that I have read the CCT policy on Academic Misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source. I declare it to be my own work and that all material from third parties has been appropriately referenced. I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution. |

SB+

- MSc in Data Analytics

Author: J. O’Brien

e-mail: [sbs23011@student.cct.ie](mailto:sbs23011@student.cct.ie)

Student ID: SBS23011

Table of Contents

[Introduction 5](#_Toc135859177)

[Assessment Process 5](#_Toc135859178)

[Data Licensing 5](#_Toc135859179)

[Business Understanding 6](#_Toc135859180)

[Data Understanding / Preparation 7](#_Toc135859181)

[Programming choice: 8](#_Toc135859182)

[Programming choice: 8](#_Toc135859183)

[Programming choice: 8](#_Toc135859184)

[Modelling 11](#_Toc135859185)

[Programming Choice: 11](#_Toc135859186)

[Programming Choice: 12](#_Toc135859187)

*The following assessment reviews publicly provided datasets from several online sources to compare the Irish construction sector with various analogous countries to determine the effectiveness/performance of this industry across several different key metrics.*

*The primary sources of data for this assessment come from both Eurostat’s key figures on European business located at* [Key figures on European business (europa.eu)](https://ec.europa.eu/eurostat/cache/htmlpub/key_figures_on_european_business_2021/index.html) *as well as from the European construction sector observatory (ECSO), an organisation that regularly carries out comparative studies against the construction sector in all 27 countries across the European Union* (Laura Hevia, 2022)*.*

*Throughout the assessment the use of Python as the programming language of choice will be executed with Jupyter notebooks used to implement all code. All source code and data files are stored in a GitHub repository located at* [jobriencct23/Semester1 (github.com)](https://github.com/jobriencct23/semester1/) *under the directory CA2, with Git utilised for version control and GitHub project features used to track issues throughout (project items) on a kanban style project board.*

## Introduction

I decided to utilise the European construction sector observatory (ECSO) and Eurostat published reports as a basis for determining datasets to use as these bodies provide in-depth analysis against the construction sector across multiple analytical and thematic reports. In addition to providing guidance for the industry the reports include specific datasets utilised from Eurostat, the statistical office of the European Union which is a body that produces European statistics in partnership with National Statistical Institutes and other national authorities in the EU Member States (Eurostat, 2020)

## Assessment Process

Throughout the assessment I utilised the guiding principles of the CRISP-DM process for my data analysis methodology implementing the following sections of this framework:

1. Business understanding
2. Data understanding
3. Data Preparation
4. Modelling

The deployment section of the process was not implemented as this is an academic and research paper not a production deployment of data models (IBM, 2021).

## Data Licensing

The Eurostat website [Home - Eurostat (europa.eu)](https://ec.europa.eu/eurostat/web/main/home) which I utilised as the location for obtaining all datasets used in this report references the legal notice of the European Commission at <https://ec.europa.eu/info/legal-notice_en> as the basis for the copyright and license details of all datasets provided. This legal notice specifies any content utilised unless specifically detailed in copyright notices is shared under the Creative Commons Attribution 4.0 International (CC BY 4.0) licence which specifies:

“Reuse is allowed, provided appropriate credit is given and changes are indicated”.

Specifically, the creative commons licensing allows for:

* Redistribution of the datasets in any medium by copying, sharing etc.
* Remixing to build upon or otherwise transforming the data for any purpose.

(Creative Commons, 2023)

## Business Understanding

Various sources were utilised in defining the choice of datasets to target for this report and included reviewing a variety of reports aimed at the construction sector including as mentioned the *key figures on European Business* report published by Eurostat that specifically targets the construction sector (Giovanni Albertone, 2021)

In addition to this report several additional reports published by the European construction sector observatory (ECSO) were reviewed as part of dataset research including:

* Stimulating favourable investment conditions
* Improving the human capital basis
* Improving energy and resource efficiency
* Housing affordability and sustainability in the EU

(ECSO, 2017 - 2021)

The final research items I utilised that assisted in deciding upon which datasets to choose included reviewing Eurostat’s short term business statistics on the construction sector, these statistics are utilised by Eurostat to closely track the business cycle of an economy using a monthly or quarterly time period for data gathering (Eurostat, 2019)

After performing this research and reviewing both the periods available and the datasets that have extensive data available as referenced by these reports, I settled on using the following datasets all sourced from Eurostat: -

|  |  |  |
| --- | --- | --- |
| **Original title** | **Eurostat location** | **Exported Version Used in Notebooks (located in CA2 data folder)** |
| Building permits | [Here](https://ec.europa.eu/eurostat/databrowser/view/STS_COBP_Q/default/table?lang=en&category=sts.sts_cons.sts_cons_per) | BuildPermitsQuarters.csv |
| House price index | Here | HousePriceIndxQuarters.csv |
| Labour input | [Here](https://ec.europa.eu/eurostat/databrowser/product/view/STS_COLB_Q?lang=en&category=sts.sts_cons.sts_cons_lab) | LabourInputQuarters.csv |
| Construction producer prices or costs, new residential buildings | [Here](https://ec.europa.eu/eurostat/databrowser/view/STS_COPI_Q/default/table?lang=en&category=sts.sts_cons.sts_cons_pri) | ProdCostsQuarters.csv |
| Production in construction | [Here](https://ec.europa.eu/eurostat/databrowser/view/STS_COPR_Q/default/table?lang=en&category=sts.sts_cons.sts_cons_pro) | ProdVolumeQuarters.csv |

Each dataset was exported from the Eurostat database and initially modified using the following methods:

* The export period set for each dataset was 2010-Q1 to 2023-Q1 to generate significant data for modelling tasks.
* Data was exported for a small, select set of countries namely (Ireland = IE, Austria = AT, Belgium = BE, Norway = NO) based on the following selection methods:
  + GDP per Capita (GDP per head of population to determine countries with closely matching purchasing power and similar population size)
  + GDP per millions

(The files used to compare countries using these GDP metrics can be found in the working folder of CA2 (GDP\_millions.xlsx and GDP\_perCapita.xlsx)

* Files were exported using the uncompressed TSV method of export which generated time series data (1 time-series = 1 row).
* Each file was then converted to CSV format prior to any data investigation or manipulation.

It is important to note that the data collected throughout this exercise is based on units of measure set from a baseline year with each subsequent calculation of a unit of measure based on an aggregate from that baseline year. In this way the units of measure do not represent “real world” values but rather abstract units of measure (Eurostat, 2021)

## Data Understanding / Preparation

(Jupyter Notebook: Data Understanding / Preparation: Section 1. Exploring Datasets)

To begin exploring the various datasets obtained from Eurostat I initially confirmed that each dataset was in the correct format i.e., the default encoding format of UTF-8 and then utilised the Pandas library to read this CSV into a dataframe. Taking an example datafile for ProdVolumeQuarters.csv I then performed various exploratory tasks such as:

* Listing the features of the dataset to explore interesting metrics.
* Reviewing for features that could be safely dropped as they represented Eurostat specific data not relevant to my exploratory requirements:
  + 'freq',
  + 'indic\_bt',
  + 'nace\_r2',
  + 's\_adj',
  + 'unit'
* Review the newly shaped dataframe which now consisted of Geography (country) and various columns representing yearly/quarterly statistics on the production volume for that country.
* For the purposes of creating a dataset with significant volumes of data over many years I manipulated the dataset using the melt function to move all the time series data into a single feature representing production volume with each observation using a unique designation feature named period that held an exclusive value of year/quarter for each country.

(Jupyter Notebook: Data Understanding / Preparation: Section 2. Cleaning Datasets)

### Programming choice:

Once I had completed those tasks above using a sample of one of the datasets, I determined that the most performant use of python would be to define a function to perform all these tasks as well as cleaning the data as required so that it is prepared to be stored in a single dataframe that can be employed as the base of all the succeeding statistics and modelling tasks.

### Programming choice:

The use of a single dataframe as a base from which to take averages and filtered subset dataframe versions, would save on memory usage and provide a single amalgamated dataframe and ease of use in visualisations, statistics tasks etc. The use of a function would reduce the need for applying repetitive code against these tasks.

(Jupyter Notebook: Data Understanding / Preparation: Section 3. Feature Engineering)

Once the various datasets were prepared the melt function was applied to each to essentially turn these datasets into new features that would be amalgamated into that single dataframe.

### Programming choice:

In this section once again, the decision was made to automate repetitive tasks for data cleaning, in this instance using a single use lambda function when required to strip out non-numeric data from specific columns.

One enhancement that could be performed against this section of feature engineering would be to create a loop structure around the tasks being performed against each original dataset, however as the code sections were easily repeated and due to time constraints, this was not implemented representing a trade-off between implementing a desired enhancement versus investing time in other areas of the data exploration.

Once the various datasets were then manipulated into long format dataframes, each one of these dataframes was then merged to create the single dataframe that would be used as the basis for the next actions.

(Jupyter Notebook: Data Understanding / Preparation: Section 4. Visualising Data)

In this next section and utilising the newly created dataframe that included country, year, and quarterly period data for constructions statistics I was able to clearly demonstrate some insights into the construction sectors by comparing countries with relatively similar GDP levels to Ireland (at least most similar when reviewing the Eurozone countries). The first graph utilised a line plot and a newly created yearly average dataframe that included the yearly average production volume for each country. Plotting a graph, I could then easily identify interesting trends in production volume over time, one such insight concluded that the production volumes for Ireland and Austria have converged in recent years to nearly identical levels. It can be seen from this graph that Irelands production volume since 2019 has drastically dropped whilst Austria’s has continued to rise until they have converged to almost identical levels by 2022.

The next visualisation I created was an interactive dashboard using plotly express and dash, I decided to create a dashboard using this method as it both allows us to run dashboards that can be exported to html / browser interfaces for consumption but also allows running the dashboard inline in a Jupyter notebook, providing both an easy way to consume whilst also documenting efficiently. The dashboard includes three pieces of data in terms of the Price / Cost / Volume of each countries construction sector, for ease of consumption I created a drop-down filter allowing reviewers to set a country and then view how the price and the costs of a production sector are correlated and how this is then reflected in the volume output for a country. One interesting insight for Ireland was that at a certain price index point (i.e., cost of housing) there seems to be a sharp corresponding increase in the cost of production across a cluster of years ranging from 2018 to 2021.

(Jupyter Notebook: Data Understanding / Preparation: Section 5. Descriptive Statistics)

As I continued to explore the newly created dataframes containing my construction data I decided to use descriptive statistics to review the data for any patterns that were evident. At first, I utilised the describe method to observe mean, median, mode and min/max values across all the features in the yearly average dataframe. I was able to observe from this table that the maximum values across each feature were significantly larger than the mean values suggesting the presence of outliers in the data.

Next, I wanted to observe if there were any strong correlations between the ProductionVolume feature and the rest of the features in the dataframe. After applying the correlation method, I was able to observe both the strongest and weakest correlations against the ProductionVolume, to further illustrate the point I used matplotlib to plot both the lowest and highest correlations, this allows me to highlight a particular feature (namely LabourInput) that has a high correlation with an increase in production volumes.

(Jupyter Notebook: Data Understanding / Preparation: Section 5. Inferential Statistics)

In this section I started out the exercise by creating a sample of my total population of construction data from the dataframe df\_constats, this was to illustrate the use of random sampling, to ensure the sample taken matches the distribution of the data included in the population I used a plot to demonstrate and compare both the original data shape and the sample shape.

The set of inferential tests included the following parametric tests:

* Two-sample T-test, the categorical variables were set as the countries of comparison (namely Austria and Ireland). The hypothesis then tested against was Null = Mean production volume is the same, Alternate = that production volume mean for Austria is less than Ireland. Using a significance level of .1 the test failed to reject the Null hypothesis.
* An additional Two-sample test was carried out for Belgium with the same result of failing to reject the null hypothesis, in one respect it is expected that production volume means would be similar as the choice of countries for this study was decided upon due to their similarities in size and GDP.
* The final parametric test was to use ANOVA testing to determine if there is a significant difference between labour input across all the 4 quarters of a year this would provide valuable information on a feature and how seasonality could affect available construction labour. A box plot was first used to visualise the differences in mean, then using pingouin to determine if the p value is smaller than our 0.2 alpha value for at least two categories; it was not. Further testing was carried out with only one p-unc value less than alpha supporting this outcome.
* For non-parametric testing a scenario was developed for testing a hypothesis of independent numeric samples namely permits issued per quarter with the Null = Austria issuing more per quarter Vs Ireland. In this instance the P – value again rejects the Null hypothesis.
* The final parametric test was to use Kruskal-wallis to investigate if there is a significant difference between issued permits (across all geographies) for at least two countries, in this instance the p-value supported this hypothesis.

## Modelling

(Jupyter Notebook: Data Understanding / Preparation: Section 5. Descriptive Statistics)

For this section I first attempted the task of analysing sentiment around the construction sector across Ireland, comparing with countries in our target datasets for the year 2023 and using the following workflow: -

* I decided to use the European Construction Industry Federation (FIEC) as the basis for my sentiment reports for each country as so far, all data utilised in this assessment has been government supplied through EUROSTAT whilst the FIEC represents employers in this sector (FIEC, 2023) and also creates reports in an additional file format.

### Programming Choice:

FIEC generates its reports in PDF, to utilise these reports for sentiment analysis I first needed to download and then use a python library PyPDF2 with the module PdfReader to be able to extract text data from this file format. I then stored results in an output json file appending to this file for each sentiment scoring round performed.

* Once the reports were identified, downloaded I was able to extract text sentiment by utilising the built in NLTK sentiment analyser Vader (Valence Aware Dictionary and Sentiment Reasoner) to perform analysis of several reports. Comparing two countries Ireland and Belgium shows a general neutrality in sentiment which is possibly due to known limitations with the built-in sentiment analyser when looking through paragraph style documents (as opposed to short form such as tweets with which the model was trained) an alternate to this would be to run another NLP classification using for example flair, the trade off to this would be that flair tends to need longer computation time to complete.

(Mogyorosi, 2020)

The next modelling task was to perform a supervised machine learning scenario using linear regression I chose this model as there are continuous values included in my target feature which will be ProductionVolume.

For the multiple linear regression scenario, the workflow was as follows:

* Assign variable X to contain all features except production volume, assign y to our target feature of production volume.
* Split the data into training and test data.
* Instantiate the model, fit, then predict.

Once these tasks were complete, I could then generate an R squared value which came to 0.7913852324981547 a relatively high value close to 1, leading me to the conclusion that variance in the target variable have a strong correlation using the features chosen for this dataframe.

In addition, I calculated the RMSE (root mean square error) to demonstrate the model’s average error for the production volumes generated for each country. A relatively low level of 7.66 per quarter for production volumes.

To visualist model performance I used a plot to show actual Vs predicted model scores

Next, I used cross validation to ensure the R squared value is accurate as we need to account for the fact that we used a sample of our data and because as observed during the descriptive statistics phase we are confident that there are significant levels of outliers in our data that could be included in these data splits. To do this I performed the following:

* Used sklearn cross validation model and K-folds.
* Shuffled the dataset.
* I used a random seed to ensure repeatability of this testing process.
* I then printed out the R squared values.

### Programming Choice:

As the number of k-folds utilised can increase execution time I tested code performance by inflating the number of splits upwards from 5 to 10, to 20 all the way up to 30 at which point the R squared value moved significantly to 0.62 approx. At this point execution of the cross-validation task went from sub 0 seconds to 0.1 (this was confirmed by installing the nbextension for Jupyter notebooks that tracks execution time). This was to demonstrate how little performance tuning this model combined with this data requires.

The final task for this section involved using the lasso regression model to determine what features have an appreciable effect on production volumes. As before the same workflow was followed to apply the model to the split test and train data, and a plot used to demonstrate the effects (both negative and positive) of certain features. To further tune this model, I utilised GridSearchCV and once the optimal alpha score was retrieved applied this to the model to plot a comparison.

In summary using regression techniques I was able to utilise my collected data to make predictions on expected production volumes and also identify the specific features that have a substantial effect on prediction models.