**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](#_Toc135677509)

[Assessment Process 5](#_Toc135677510)

[Data Licensing 5](#_Toc135677511)

[Business Understanding 6](#_Toc135677512)

[Data Understanding / Preparation 8](#_Toc135677513)

[Data Understanding / EDA 11](#_Toc135677514)

[Using a histogram (Volume of applications) 11](#_Toc135677515)

[Using a line plot (Lead time progression) 11](#_Toc135677516)

[Using a histogram (Outliers and measure of centre) 11](#_Toc135677517)

[Data Understanding / Poisson Distribution 12](#_Toc135677518)

[Code Choice 13](#_Toc135677519)

[Result 13](#_Toc135677520)

[Data Understanding / Normal Distribution 14](#_Toc135677521)

[Modelling / Machine Learning 14](#_Toc135677522)

[Classification 15](#_Toc135677523)

[Code Choice 15](#_Toc135677524)

[Result 16](#_Toc135677525)

[Regression 17](#_Toc135677526)

[Code Choice 17](#_Toc135677527)

[Result 18](#_Toc135677528)

[Programming Paradigm 18](#_Toc135677529)

[Investigation of R 19](#_Toc135677530)

[In Summary 20](#_Toc135677531)

[References 1](#_Toc135677532)

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

*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 organsiation 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 (project items).*

## Introduction

I decided to utilise the European construction sector observatory (ECSO) and Eurostat published reports as a basis for determining datasets to utilise as these bodies provide in-depth analysis against the construction sector in several 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, n.d.)

## Assessment Process

Throughout the assessment I utilised the guiding principles of the CRISP-DM process for the project approach implementing the following sections of this methodology:

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

The deployment section of the process however was not implemented as this is an academic and research paper (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, n.d.)

## 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 close 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 i.e. 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, n.d.)

## 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 a unique 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 utilised as the base of all the subsequent statistics and modelling tasks. This would save on memory usage and provide a single amalgamated dataframe for use in visualisations. 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 a 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 could 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 dataset that would represent the data for my modelling and statistical tasks to come (as well as making it easier to visualise data).

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

In this next section and utilising the newly created dataframe including country, year, and quarterly period data for constructions statistics I was able to clearly demonstrate some insights into the construction sectors for our comparison 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. Use 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 similar 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 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 variable 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 the previous 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 permits (across all geographies) for at least two countries, in this instance the p-value supported this hypothesis.

## Data Understanding / EDA

Once the data was cleaned and data sets merged in a single data frame, the data was explored using a variety of visualisation techniques to better understand the provided data.

(Jupyter Notebook: Data Understanding / EDA: Section 1. Visualisations)

### Using a histogram (Volume of applications)

A histogram was utilised to demonstrate the overall spread of applications across various postcodes, the planning applications per postcode were demonstrated in this diagram to be centred mainly around ten postcodes with some of the areas representing a very small number of overall applications. The most popular location is Dublin 1 (D1) and using this graph plot D1 was identified for use in the descriptive statistics scenarios (see Poisson Distribution etc. below).

### Using a line plot (Lead time progression)

As the data frame now included data over time, outcomes, and a breakdown of how long an application took this data could be plotted in a line plot and clearly demonstrate patterns and observations in the data that may warrant future investigation. Observations including:

* For the year 2005 the number of successful and unsuccessful applications converged with almost an even number of failures to successes being observed.
* This trend was almost reversed in the year(s) 2020 to 2021 as successful applications were much higher relative to unsuccessful (though the unsuccessful rate was relatively static and there were far more successes than failures).

### Using a histogram (Outliers and measure of centre)

(Jupyter Notebook: Data Understanding / EDA: Section 2. Descriptive Statistics)

The mode for the categorical nominal data supplied in the DECISION feature determines that overall planning applications are more often successful than not based on historical applications.

Utilising the newly created quantitative feature of LEADTIME it was possible to generate the mean, for both individual locations and overall measures of centre to determine how long an application took going from application to decision on average (Cotton, 2022).

This was based on a total size of 71117 applications over 20 years resulting in:

mean 58.300443

whilst the median equalled:

52.000000

We could also see that there was a max outlier of 1652 for a planning application to achieve decision, and results like this are likely to have a major effect on the overall resultant mean.

To investigate this with one specific area, namely Dublin 1 (D1) using the mean and median of:

mean 62.087957

median 53.000000

A histogram was then utilised to plot all lead times showing the spread of data as asymmetrical in that it is right skewed data. The supposition is that major outliers are influencing the mean and median. To view this more clearly another histogram was utilised to exclude any extreme outliers by using values up to 100 days only for an application. This clearly shows that it is far preferable to use the median than the mean in future analysis efforts as it is less affected by outliers. (Matsui, 2020)

## Data Understanding / Poisson Distribution

(Jupyter Notebook: Data Understanding / EDA: Section 3. Poisson Distribution)

As part of analysing the supplied data, it was decided that a Poisson distribution would be used to provide insights into the potential success rate of planning applications. The choice of Poisson distribution is justified as we have a random number of independent outcomes i.e., the planning applications are not limited or specific in that we can look at the average numbers per years and weeks but there is no definitive limit on applications, so the sample is random. (Matsui, 2020)

This precluded the use of binomial distribution even though the outcome is one of two outcomes i.e., successful, or unsuccessful to be suitable we would require a definitive limit on the number of applications which is something that cannot be specified using the supplied data.

Utilising this distribution, it was possible to develop a scenario based on the following parameters:

* Taking any one specific location i.e., Dublin postcode
* Looking at the number of applications on average per week for that postcode (this would represent the lambda value)
* Develop scenarios to define the likelihood of values over and above the average range of planning applications per week.
* This data could be utilised in many scenarios such as defining staffing rates per week, evaluating turnaround times for applications etc.

### Code Choice

Python was utilised to calculate the average number of applications per week from a specific location. The resulting average was then utilised, and Poisson imported fromSciPy. Stats to be utilised for predicting the likelihood of planning applications per week reaching certain values (Zach, 2021).

### Result

The following scenarios were tested with Poisson Distribution:

* What is the probability of receiving 10 planning applications in a week?

0.07098326865041356

* What is the probability of receiving 3 planning applications in a week?

*0.052129252364199796*

* What is the probability of receiving more than 15 planning applications in a week?

*0.005717202492495965*

Using these types of scenarios, we could set upper and lower boundaries for the likelihood of having certain numbers of planning applications per week and thus set appropriate performance indicators for processing, staffing numbers or any other number of useful inferences from the data supplied.

## Data Understanding / Normal Distribution

(Jupyter Notebook: Data Understanding / EDA: Section 4. Normal Distribution)

It was observed that whilst plotting the Poisson distribution against planning application lead times using low sample sizes did not result in anything resembling a normal distribution i.e., the curve is not symmetric. (Matsui, 2020)

The gaussian distribution only appears when generating a very high sample size of 1000. Once implemented a normal distribution shape appears supporting the expectation of central limit theorem, meaning that as expected large sampling sizes leads to the normal shape. (Matsui, 2020)

The ramifications for lead time data being “non normal” suggest from perusing the data that many outliers were causing skewed results, and as the application process has no set min / max limits for applications this data can wildly affect distribution (SOLVERMARK, 2010) . This hypothesis was tested using a scatter plot which clearly highlights wild swings in terms of lead time outliers with orders of magnitude far beyond the mean or median.

## Modelling / Machine Learning

Once the data had been examined and evaluated using visualisations and discrete statistical analysis techniques, several scenarios for machine learning scenarios were evaluated based on the amount of data available, various feature types available (and created using feature engineering) as well as the type of data available.

As the supplied data sets included either existing labelled features or in some instances newly created and labelled features the decision was made to utilise supervised learning models. In addition, as the datasets included a mix of both numeric and categorical data it was possible to envisage scenarios for both binary classification (predict one of two outcomes) as well as predicting continuous values using regression techniques (Ali, 2022).

To satisfy the requirements of supervised learning in the data preparation phase all data was stored in data frames or NumPy arrays, it was ensured no observations contained missing data and data were converted to numeric as needed (Boorman, n.d.).

### Classification

The classification scenario was determined as follows:

“Use a machine learning algorithm and available data features to predict the outcome of the DCC planning applications process, as: successful or unsuccessful”.

Two machine learning algorithms were trained for classification:

* Decision Tree
* K-NN (K nearest neighbour)

### Code Choice

For both classification models the sci-kit learn workflow was utilised in that a model is imported, variables created then instantiated for training and test data sets to ensure adequate performance, the model is fit to the data and then hyperparameters are evaluated to ensure the best prediction accuracy.

#### Decision Tree

(Jupyter Notebook: Data Modelling: Section 1. Classification with Decision Tree)

This algorithm uses a sequence of if-else questions to infer labels (outcomes) and was implemented using the sci-kit learn DecisionTreeClassifier module (Kawerk, n.d.). The scenario devised used a data frame that include a subset of applications for the D1 location by creating a data frame filtered on that location name only. The *YEAR* and *LEADTIME* features were utilised, placed into variables for X and then y was set to the *DECISION* feature.

Other required actions included splitting the data into training and test parts using an 80/20 split as well as using the StandardScaler function to ensure that the error and accuracy rates are not biased due to the differing scales of the numeric data dimensions utilised (Mulani, 2022), the scalar changes these data values into a standard format with a mean as 0 and standard deviation as 1.

To tune model performance the use of GridSearchCV (cross validation) was implemented using an array of hyperparameters including max\_depth which determines the maximum number of branches utilised. In this instance the use of 6 gave an optimised accuracy score (Boorman, n.d.).

To plot and visualise the accuracy of the model a heatmap / confusion matrix (Kundu, 2023) is utilised and for our two features demonstrates the distribution across true / false positive and negative outcomes, in other words how often classifications are correctly assigned and of import is the diagonal of the heatmap which shows True Positive / True Negative outcomes for the model.

#### K Nearest Neighbour

(Jupyter Notebook: Data Modelling: Section 2. Classification with K-NN)

The second classification algorithm utilised was K-NN where K represents the number of closest labelled data points, which then uses majority voting on what the unlabelled observation should be. In other words, the prediction is based on the labels that most of the nearest neighbours have.

As before training and testing data are utilised and placed in arrays for future usage, to tune performance (in other words to find the best value for K) an array of 1 to 20 was generated and a for loop utilised to iterate through this array. To evaluate the best K value two methods were utilised to demonstrate and visualise the best K value:

One method is to plot the error rate against K values in the array, showing a graph demonstrating the lowest error rate obtained by K value (Band, 2020). The alternate way was to invert this concept and instead show K value to accuracy rate compared with both the testing and training data. Both plots have been utilised in the Jupyter notebook but essentially show the same information using differing formats for plotting.

The outcomes of the proposed K value were then plotted to show testing and training accuracy levels in a graph format, the intersection of these values representing the K value providing the most accurate score.

As per the previous method a confusion matrix is plotted to show True Positive / True Negative along with the overall accuracy of the model.

### Result

Use of both Decision Tree and K-NN lead to similar accuracy rates (Allwright, 2022), to validate this score as useful for our prediction purposes, in python and using pandas the successful to unsuccessful values are calculated with a 60 / 40 split indicating that the data we are working with is “balanced” (Allwright, 2022) and with the accuracy rate supplied the use of either model is adequate for predicting planning application outcomes.

### Regression

The regression scenario was determined as follows:

“Use a machine learning algorithm to predict the outcome of the Dublin City Council planning applications process, with respect to the expected number of successful applications in a given year based on the total number of applications submitted for that year”.

To utilise a Regression classification model, it was required to create another continuous variable that could be utilised specifically for a Simple Linear Regression. In this way there would be both a dependent and independent variable to utilise for the regression.

### Code Choice

To create the required variable and feature category a new data frame was created using the full set of data. Two new features were created that included a total count of all applications submitted grouped by year with all successful applications in another feature. This was implemented using a small function (lambda), an additional column was then utilised to hold all successful decisions.

This new data frame would then be utilised for the linear regression model. In addition, as opposed to using sci-kit learn statsmodel was utilised and specifically OLS was used to fit a model to the new data frame. (Broeck, n.d.)

With the new data frame created and two continuous features available the relationship between a response variable (number of applications) and the explanatory variable (number of successful applications) could be explored. In other words, given a certain number of planning applications per year, how many successful applications can we predict will occur.

First a correlation was calculated and shows a strong positive correlation of 0.97, in other words as the number of applications increase the number of successful applications typically increases in tandem. To demonstrate this insight a scatter plot was utilised that clearly demonstrates the correlation and a trend line added using *regplot,* this trend lineclearly demonstrates the plotted linear regression line close to the data points (this is highlighted also with the use of a confidence interval outline).

To run the linear regression, model the ordinary least squares regression is utilised with two arguments passed to this function, explanatory variable to the right and response variable to the left. *DotFit* is used to fit the model.

To obtain predictions a pandas data frame is created to store a range of values representing the number of applications submitted. To then measure the performance of this model the r-squared value was calculated using rsquared (Broeck, n.d.).

The r-squared value measures between 0 and 1, with 1 representing a perfect score. This value represents the coefficient of determination which is calculated as the correlation squared. This value represents the proportion of the variance in the response variable that is predictable from the explanatory variable.

In addition, another model performance metric utilised was the residual standard error, which represents the typical difference between a prediction and an observed response. To obtain the RSE it needed to be taken from another related but less commonly used metric that of the MSE (mean squared error), this metric is in fact the RSE squared so to retrieve the RSE first the MSE is found and then the square root of this value taken. (Broeck, n.d.)

### Result

We used this model to predict how many of the applications are expected to be successful given an array of predictions to make, the model was then evaluated using two metrics commonly utilised for measuring linear regression to determine model performance.

## Programming Paradigm

Python is an object-oriented programming language introduced in 1989 and is considered general purpose, easy to read and learn. Python libraries and packages are utilised throughout this assessment including:

* Numpy (Numerical Python)

Used for working with arrays and provides a large library of mathematical functions to work with these arrays, particularly useful for visualisation efforts (Council, n.d.).

* Pandas

Used to for data manipulation and analysis (Schools, n.d.).

* Numpy

Used for performing mathematical operations on arrays and matrices.

(Training, n.d.)

* Seaborn and Matplotlib

For plotting and visualisation of data.

* Datetime

A module for manipulating dates and times.

* Math

This module is used for basic and advanced mathematical operations using a variety of functions (Liyanapathirana, n.d.).

* Scipy.stats

Utilised for a variety of statistical functions and in the process of descriptive statistics.

* Scikit-learn

Machine learning algorithm utilised in regression and classification scenarios.

### Investigation of R

An alternative to Python for many data analysts when looking at investigating and manipulating data is the R programming language. Whilst python is considered by some indexes the most popular language by market share R has alternated over the last number of years from number 8 to outside the top 20 on various popular ranking indexes.

Though there are several similarities between both programming languages including their open-source nature, and strong communities there are some significant differences that would potentially allow for the choice of R for this assessment over python and vice versa (Ozgur, 2008).

#### Adoption Considerations

One of the key factors when choosing a language for any task will be the skill and specialisation requirements to learn the language. Whilst R is considered primarily a language abstracted for use specifically in areas like academia and easy to use whilst starting out, complexity is quite often introduced rapidly for more advanced tasks. Python on the other hand is considered one of the closest programming languages to natural English language and its syntax emphasises ease of use and adoption.

#### Data Analysis Goals

A key difference with the development of R over Python is that whilst Python is a general-purpose language that has been developed without a specific focus on data analysis it is nevertheless a very popularly utilised language for this purpose and is bolstered for this task using libraries and tools that provide this functionality to python.

R on the other hand is very often utilised by statisticians as its development is very much focused on specialised analytics and stat models. Often just a few lines of code suffice to generate to implement deep statistical analysis of datasets (Ozgur, 2008).

#### Data Collection

Another key difference between python and R is that whilst python can ingest from several data file types including CSV, JSON, and directly from SQL tables, web sites etc. R On the other hand is specifically designed for importing CSV, excel and text files. There are additional packages like Rvest used to help with things like web scraping, but it is considered more extensive on the Python side (Team, 2021).

#### Data Visualisation

Whilst Python has several libraries for generating graphs and other visualisations one of the specific goals of R was to demonstrate the results of statistical analysis using rich and advanced graphics implementations (Ozgur, 2008).

#### Ecosystem

Whereas python has the far larger number of packages (300,000 compared with 19,000 for R) it should be noted that as it is focused on many different programming domains many of these packages are not utilised for data analysis. R on the other hand is focused on data science though this could also be considered a downside as it is difficult at times to find the right package (Luna, 2022).

### In Summary

Whilst the use of Python for data analysis is a popular and valid choice due to its flexibility and rich supported ecosystem of libraries and packages, the alternative viewpoint is that R whilst not as popular, or easy to learn is and was developed specifically with the goal of providing a rich toolset for statistical and data analysis using best-in-class graphics and visualisations. Another viewpoint is that an argument can be supported for combining both Python and R depending on the specific requirement and objectives of any analysis using an IDE like RStudio which supports combined use of both languages (McNulty, 2021).

# References

@programizstudios,2021.*Python strftime().*

[Online]   
Available at:

https://www.programiz.com/python-programming/datetime/strftime  
[Accessed march 2023].

Ali,M.,2022.*SupervisedMachineLearning.*

[Online]   
Available at:

https://www.datacamp.com/blog/supervised-machine-learning  
[Accessed March 2023].

Allwright,S.,2022.

*What is a good accuracy score in machine learning?.*

[Online]   
Available at:

https://stephenallwright.com/good-accuracy-score/#:~:text=There%20is%20a%20general%20rule%20when%20it%20comes,70%25%20-%20OK%204%20Below%2060%25%20-%20Poor  
[Accessed March 2023].

Allwright,S., 2022. *What is imbalanced data? Simply explained.*

[Online]   
Available at:

https://stephenallwright.com/imbalanced-data/  
[Accessed April 2023].

Azevedo, A. a. S. M., 2008. *KDD, SEMMA and CRISP-DM: a parallel overview.*

[Online]   
Available at:

http://hdl.handle.net/10400.22/136  
[Accessed March 2023].

Band, A., 2020. *How to find the optimal value of K in KNN?.*

[Online]   
Available at:

https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb  
[Accessed March 2023].

Boorman, G., n.d. *Machine learning with scikit-learn.*

[Online]   
Available at:

https://campus.datacamp.com/courses/supervised-learning-with-scikit-learn/classification-1?ex=1  
[Accessed April 2023].

Broeck, M. V. D., n.d. *Introduction to Regression with statsmodels in Python.*

[Online]   
Available at:

https://app.datacamp.com/learn/courses/introduction-to-regression-with-statsmodels-in-python  
[Accessed April 2023].

Broeck, M. V. d., n.d. *Simple Linear Regression Modelling.*

[Online]   
Available at:

https://campus.datacamp.com/courses/introduction-to-regression-with-statsmodels-in-python/assessing-model-fit-e78fd9fe-6303-4048-8748-33b19c4222fe?ex=1  
[Accessed March 2023].

Bush, T., 2020. *Descriptive Analysis: How-To, Types, Examples.*

[Online]   
Available at:

https://pestleanalysis.com/descriptive-analysis/#:~:text=Descriptive%20analysis%2C%20also%20known%20as%20descriptive%20analytics%20or,to%20generate%20accessible%20insights%20from%20otherwise%20uninterpreted%20data.  
[Accessed March 2023].

Cotton, R., 2022. *Data Demystified: An Overview of Descriptive Statistics.*

[Online]   
Available at:

https://www.datacamp.com/blog/data-demystified-an-overview-of-descriptive-statistics  
[Accessed April 2023].

Council, D. C., 2023. *Planning Applications DCC.*

[Online]   
Available at:

https://data.gov.ie/dataset/dublin-city-council-planning-applications  
[Accessed March 2023].

Council, D. C., 2023. *Residential Planning.*

[Online]   
Available at:

https://www.dublincity.ie/residential/planning  
[Accessed March 2023].

Council, N. S., n.d. *Numpy.*

[Online]   
Available at:

https://numpy.org/  
[Accessed March 2023].

Dost Muhammad Khan, N. M., 2013. *A Unified Theoretical Framework for Data Mining.*

[Online]   
Available at:

https://www.sciencedirect.com/science/article/pii/S1877050913001488?ref=pdf\_download&fr=RR-2&rr=7b63d772783c240b  
[Accessed March 2023].

GeeksforGeeks, 2023. *OrderedDict in Python.*

[Online]   
Available at:

https://www.geeksforgeeks.org/ordereddict-in-python/  
[Accessed April 2023].

Kawerk, E., n.d. *Decision Tree for Classification.*

[Online]   
Available at: https://campus.datacamp.com/courses/machine-learning-with-tree-based-models-in-python/classification-and-regression-trees?ex=1  
[Accessed April 2023].

Kelley, K., 2023. *What is Data Analysis? Methods, Process and Types Explained.*

[Online]   
Available at:

https://www.simplilearn.com/data-analysis-methods-process-types-article  
[Accessed 03 2023].

Kundu, R., 2023. *Confusion Matrix: How To Use It & Interpret Results.*

[Online]   
Available at:

https://www.v7labs.com/blog/confusion-matrix-guide  
[Accessed March 2023].

Liyanapathirana, L., n.d. *The Python math Module: Everything You Need to Know.*

[Online]   
Available at:

https://realpython.com/python-math-module/  
[Accessed March 2023].

Luna, J. C., 2022. *Python vs R for Data Science: Which Should You Learn?.*

[Online]   
Available at:

https://www.datacamp.com/blog/python-vs-r-for-data-science-whats-the-difference  
[Accessed April 2023].

Matsui, M., 2020. *Introduction to Statistics in Python.*

[Online]   
Available at:

https://app.datacamp.com/learn/courses/introduction-to-statistics-in-python  
[Accessed April 2023].

Matthes, E., 2019. Python Crash Course. In: *Python Crash Course.* s.l.:s.n., pp. 337-339.

Matthes, E., 2019. Python Crash Course. In: *Python Crash Course.* s.l.:No Starch Press, pp. 49 - 70.

McNulty, K., 2021. *Five ways to work seamlessly between R and Python in the same project.*

[Online]   
Available at:

https://towardsdatascience.com/five-ways-to-work-seamlessly-between-r-and-python-in-the-same-project-bf173e35fdef  
[Accessed March 2023].

Mulani, S., 2022. *Using StandardScaler() Function to Standardize Python Data.*

[Online]   
Available at:

https://www.digitalocean.com/community/tutorials/standardscaler-function-in-python  
[Accessed April 2023].

Ozgur, C. C. T. R. G. a. H. Z., 2008. MatLab vs. Python vs. R.. *Journal of data Science,* 15(3), pp. 355-371.

Pydata.Org, n.d. *https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.merge.html.*

[Online]   
Available at:

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.merge.html  
[Accessed March 2023].

Schools, W., n.d. *Pandas Tutorial.*

[Online]   
Available at:

https://www.w3schools.com/python/pandas/default.asp  
[Accessed March 2023].

SOLVERMARK, 2010. *The 7 Biggest Reasons That Your Data Is Not Normally Distributed.*

[Online]   
Available at:

https://discover.hubpages.com/education/reasons-that-your-data-is-not-normally-distributed  
[Accessed October 2023].

Team, I. C., 2021. *Python vs. R: What’s the Difference?.*

[Online]   
Available at:

https://www.ibm.com/cloud/blog/python-vs-r  
[Accessed March 2023].

Training, D. F., n.d. *NumPy Applications – Uses of Numpy.*

[Online]   
Available at:

https://data-flair.training/blogs/numpy-applications/  
[Accessed April 2023].

W3, 2022. *Pandas DataFrame iterrows() Method.*

[Online]   
Available at:

https://www.w3schools.com/python/pandas/ref\_df\_iterrows.asp  
[Accessed March 2023].

ZACH, B., 2021. *How to Use Pandas fillna() to Replace NaN Values.*

[Online]   
Available at:

https://www.statology.org/pandas-fillna/#:~:text=You%20can%20use%20the%20fillna%28%29function%20to%20replace%20NaN,%23replace%20NaN%20values%20in%20one%20columndf%5B%27col1%27%5D%20%3D%20df%5B%27col1%27%5D.fillna%280%29

[Accessed March 2023].

Zach, B., 2021. *How to Use the Poisson Distribution in Python.*

[Online]   
Available at: https://www.statology.org/poisson-distribution-python/