A picture containing icon

Description automatically generated

**CCT College Dublin Continuous Assessment**

**Abstract**

*Inbound tourism is a huge part of the Irish economy with almost 8.5 million visitors recorded in 2008 (Callaghan and Tol, 2011). Critically analysing the number of people who visit and who leave, how they travel to and from the country and the time of their visit was the aim of this work. EDA was done for data preparation for the statistical analysis and machine learning algorithms that were applied to the dataset for insight into passengers' movement patterns. The information gleaned from these processes was then used to infer future trends. Models were applied to predict if the type of movement i.e. arriving or leaving the country could be classified accurately.*

**Introduction**

Multiple open-source resources were explored to find a dataset containing relevant data regarding the Irish tourism industry. After consideration, the dataset was downloaded from data.gove.ie (Mahon, 2021), and uploaded into a pandas data frame using a Python Jupiter Notebook.

# Data preparation and Visualization

EDA aims to allow the analyst to gather an insight into the particular data set, the type of data, if there is missing data etc. and to observe potential internal patterns. It allows the analyst to become familiar with the context in which the data is set (Behrens, 1997).

The data dictionary containing the column names and units of measurement associated with the data was viewed in Excel. Duplicate data can pose a problem to the data set integrity and introduce bias into future analysis (Chu and Ilyas, 2016). It was found that the dataset didn’t contain duplicated rows. This is a relevant and observable problem, particularly in large data sets. Ananthakrishna et al. successfully employed an algorithm which handled missing data in a hierarchical structure in data warehouses (Ananthakrishna et al., 2002) .

The original data frame shape was found to have 12936 rows and 10 columns. It was found that there were 2 missing values from the passenger movement column. Due to the relatively small number of rows missing, 0.015%, it was decided to remove those rows from the dataset. If the missing data was more significant, several methods could be used such as imputation and interpolation (Bonaccorso, 2017; Kumar, 2019).

The unique values in each column were examined. It was seen that the direction was either arriving or departing and there were 11 unique values in the Type of Passenger Movement column. The data frame was filtered to select rows with common values. Sub-data frames of interest based on the direction of travel or season were saved. Care was taken not to double count passenger numbers and only relevant values were filtered for.

Previously discussed methods of exploring and understanding the data informed the decision that scaling and encoding would be required before machine learning could be carried out successfully.

Encoding is the process of assigning a numeric value to categorical data, usually in the form of a string (Bonaccorso, 2017). One-hot and label encoding was performed. Label encoding assigns ordinal values on nominal data which can be misleading to ML algorithms which cannot differentiate the arbitrary values. For instance ‘Arriving’ was assigned 0 and ‘Departing’ to 1. To the algorithm ‘Departing’ has a higher value which introduces biases. It also introduced the concept of multiple categories summing to give a third category, see Table 01 (Bonaccorso, 2017). One-hot created an additional column for each category in the data frame and assigned each a binary value, 1 if the row contained the categorical variable, otherwise 0.

The KNN algorithm used a label-encoded data frame and the other algorithms the one-hot encoded. This was because KNN became unstable due to the higher dimensionality of the one-hot encoded data frame (Kirk, 2017).

|  |  |
| --- | --- |
| Categorical Value | Encoded label |
| Passenger Movement Cross-Border Rail | 1 |
| + | |
| Passenger Movement Cross-Border Bus | 0 |
| + | |
| Passenger Movement by Air from All Airports | 2 |
| = | |
| Passenger Movement by Sea to All Countries | 3 |

Table 01: Issue which arose when encoding nominal data

The lowest passenger movement value was 4,815 and the highest was 1,603,381. As it had been established the data was skewed therefore min-max scaling was used, see line 111. Standard scaling and L2 scaling were also trialled and compared as seen in Figure 01. Both min-max and standard scaling retained the data’s shape. L2 didn’t work, this may be due to the large number of outliers within the data set. The decimal\_year values were also scaled for consistency.

A group of graphs showing different values

Description automatically generated with medium confidence

Figure 01: Comparison of scaling techniques

Year\_month\_to\_decimal and season functions were employed to prepare the dataset for ML. The seasonal trends could be seen clearly from the scatter plots. Every summer (June, July, August) the numbers arriving and departing peaked. To be able to compare like with like in ML and statistic models made this an important categorisation.

In Figure 03, 4 boxplots, the key features of this graph have been labelled. This allows someone unfamiliar with this kind of visualisation to gain an understanding easily. All of the graphs contained a colour-coded legend on the image. The legend was placed away from data points and labelled concisely to reduce size (Tufte, 2001).

The axes on the graphs were proportional to the data set. This is a key principle because if the scale of the data is not maintained the data can appear very differently to the observer. To make this visually clearer a grid was added to all the graphs. Looking at the y-axis in Figure 02, each horizontal line demarks 200 counts of an interval. Although this reduced the data to ink ratio it was decided it added more clarity than distortion. They were made light grey and thin to minimise the impact. The heatmaps shown in Figure 10 were given the same colour scale so that the models could be fairly compared (Tufte, 2001).

The style of graphs and colour scheme was kept constant to fulfil the principle of graphical integrity. Individual graphs required different stylings i.e. the bin width in the histograms, see Figure 01. All other charts had common bar width (Tufte, 2001).

The dimensions of the visualisations never exceeded the dimensions of the displayed data. For example, a scatter plot with two variables has two axes as shown in Figure 03. Effort was made to reduce space on the graph. For example, in Figure 07 an x and y limit were implemented after it was established where the data occurred, see lines 133 and 138. The aspect ratio was kept to 3:2 horizontal to vertical for all figures. No unnecessary information was added that did not contribute to the overall display of the data (Tufte, 2001).

**Statistics**

The Python pandas library was utilised to maximise efficiency and accuracy. The .dtypes function was used to determine the data types that were contained in the data frame on line 6. The number of passengers, with column name VALUE, was numeric, as was the decimal year. The direction, type of movement and month were all categorical. There were 11 types of passenger movement recorded in the dataset. The categorical data was all nominal, with no variable being superior to another. The direction could be considered a dichotomous nominal variable as it had only two unique values, arriving to or departing from Ireland. The number of passengers entering/leaving the country was a discrete numerical value. The value could also be classified as a ratio variable as it has a definite zero point.

The measures of central tendency were taken for the dataset next. This gave a mean value of 130228 with a standard deviation of 224501, indicating the data was skewed. The lower quartile, median and upper quartile values were 28723, 44200 and 124587 respectively. The mode does not apply to numeric data and did not reveal anything about the type of passenger movement as there was an equal number of entries for each category.

A green line graph with a line graph

Description automatically generated with medium confidence

Figure 02: Histogram of passenger\_movement\_net\_df

In Figure 02 the data is observed as right skewed. A KDE was overlayed onto the plot to give an estimate for the probability density function. Histograms play a key role in determining the physical layout of continuous variables (Chen et al., 2008).

It was decided to take subsets of the dataset to create more meaningful plots, statistics and probability distributions. Boxplots are an important tool to highlight outliers in a dataset (Chen et al., 2008). Figure 03 shows a side-by-side of the box plots for those arriving and departing by air and sea.

A graph of different colored squares

Description automatically generated

Outliers

Upper Quartile

Median

IQR

Lower Quartile

Figure 03: Comparison of Boxplots

The data frame was filtered to show data about those who arrived and departed by sea and air across the period. Taking a look at those arriving the two data frames (blue and yellow on the plot) have comparable medians of 177184 and 93685, denoted by the horizontal line in the box, it is clear that the two datasets were spread out over very different ranges. The whiskers on the boxplot are used to classify a point as being an outlier or not. They are calculated by multiplying the IQR by 1.5 (Chen et al., 2008). Values which lie outside of this are considered to be outliers and are denoted by a dot in Figure 03. While sea travel has remained relatively steady the aviation sector has increased significantly. The IQR of the yellow plot is 40,590 and the blue is 117,161. This measure of spread indicated that there is much more variability within the air than sea travel sector.

A graph of a graph showing the growth of a company

Description automatically generated with medium confidence

Figure 04: Scatter plot from 1961 to 2010

The third chosen plot was a scatter plot, shown in Figure 04, plotted on line 29 of the notebook. The year is plotted along the x-axis and the number of passengers moving is along the y-axis. Up until the 1990s, the data remained relatively constant with a peak occurring in each year corresponding to the summer months followed by a trough in the winter. After 1990 air travel starts to increase considerably. This reflects what was observed in the boxplot with the large variation in data for air travel.

Probability distributions allow a numeric value to be assigned to an event occurring within a given context. It is crucial to apply a relevant theory as the underlying assumptions will dictate the accuracy of the probability results. Probabilistic systems are based on the uncertain nature of events. They aim to interpret the range of possible outcomes, as opposed to the single outcome in deterministic systems given certain initial conditions (Weiss, 2012).

The binomial distribution is a series of independent trials. Due to the large number of people who entered and left over the 50-year time frame it was assumed that the law of large numbers applied. This was done in the hope that no one event had a lasting impact on the data collected such as an abnormal weather event which would influence international travel (Weiss, 2012). The probability also must remain constant for each trial. Sea travel was relatively constant both for departing and arriving. This key feature meant that it was more suitable for a binomial distribution than the entire dataset. This section was coded in lines 33 to 40 of the code.

There were three ways to enter the country, sea, air or cross-border travel, and since the binomial distribution requires only two outcomes a different question was posed.

The chosen event to explore was if a passenger travelling by sea was chosen at random what was the probability that the individual was arriving or leaving Ireland. To construct the distribution the numbers arriving into and leaving the country were summed separately and divided by the total number of sea travellers.

The pmf, sf and cdf methods were used to evaluate the probabilities over the 10 trials. The x-axis shows the number of successes out of 10 trials and the y-axis is the probability that the given person was departing from Ireland. Due to p ≈ q the distributions are symmetric. The results of these can be seen in Figure 05.

A graph of different colored lines

Description automatically generated

Figure 05: Binomial Distribution for the probability of passengers departing Ireland, n = 10

The expected value was also computed which gives the average value over the long term. Over the long term, 4.99 ≈ 5 in every 10 randomly chosen sea travellers would be departing from the country.

**A graph showing a number of trails

Description automatically generated**

Figure 06: Probability mass function for a large number of Bernoulli trials with half of the trials being a success

The probability for a large number of trials half of which resulted in a success was plotted in Figure 06. This shows that for a large n, the probability tends towards 0. This follows the characteristic that the binomial distribution can be approximated by a normal distribution as n increases and p remains fixed. This is important as it is much more computationally expensive to compute binomial distributions with large n values than normal distributions (Weiss, 2012).

Next, a Poisson distribution was explored, see lines 43 to 46. The total number of people moving by sea was calculated and divided by the total number of months. Various time frames were calculated and the probability that that number of people arrived was found and plotted. Again as λ tended towards large values the probability tended to 0, as shown in Figure 07. This was the expected result because as the time frame is increased the distribution becomes closely concentrated around the mean and any variation in it leads to a large reduction in the probability. This peak around lambda is in part reason why it too can be approximated with a normal distribution curve along with the central limit theory.

A graph with a purple line

Description automatically generated

Figure 07: Poisson distribution over a long time frame

A supplementary categorical value was added to the dataset, season, to create a subset of the data which was normally distributed. Figure 08 shows the sub-selection of data chosen with a randomly generated normally distributed curve with the same mean and standard deviation as the dataset superimposed onto it.

A graph of a graph

Description automatically generated with medium confidence

Figure 08: Validating the shape Gaussian of the dataset

The shape of the two are similar and so it was decided to apply a Gaussian theory to this dataset. The dataset was made up of passengers departing by sea during the summer. One of the most noticeable differences between this type of distribution as opposed to the two previously discussed is that normal distributions have a probability density instead of a probability mass function. This means that normal distributions give a probability over a range of continuous variable values rather than an exact probability at a given discrete random variable value (Weiss, 2012).

Some exploratory calculations were performed on the approximate normal data. The probability of between 42780 and 392103 people departing by sea during the summer is 95.44%.

The variable chosen was done so as it was a numeric value influenced by other features. The same variable was used for the discrete and continuous distributions in this case. This was possible as the distributions were measuring the probabilities of different things. The discrete distributions were finding the probability that a person was arriving or departing whereas the continuous distribution related to the probability a certain number of people would enter. This is an important distinction. Not all variables are suitable for both, categorical data cannot be modelled using a normal distribution.

1410

**Machine learning for Data Analytics**

The cross-industry-standard-process model for Data Mining (CRISP-DM) is a widely used data management framework. Many data projects do not follow a specific framework but doing so can reduce costs and completion timelines and promote best practice implementation. It has been applied to a wide range of data projects dealing with biological processes, software projects, cyber forensics and healthcare (Ayele, 2020) and it has been described as industry-independent. This indicates it has a high degree of flexibility (Schröer et al., 2021).

It can be divided into 6 main processes: Business and data understanding, Data preparation, Modelling, Evaluation and Deployment. These pillars of the framework offer a data team structure and ensure that a data science team logically approaches the problem (Schröer et al., 2021).

The shortcomings were discussed by Studer et al. CRISP-DM is focused on static analysis and is not set up to handle real-time data. This limits the dynamic decision-making capabilities of this approach. Furthermore, there is a gap in the framework relating to standards and quality assurance. As with all machine learning results the model must be assessed within their given context (Studer et al., 2021).

Machine learning (ML) algorithms are a mathematical approach to identifying patterns in noisy data. Some commonly used supervised ML algorithms were applied to this dataset to be able to predict future behaviour for passengers entering and leaving Ireland (Kirk, 2017).

Initially, the KNN algorithm was run on the dataset. The code can be found in lines 61 to 67 of the notebook. This groups data points with other nearby ones and then categorises the target variable based on which group it is nearest to. The target variable was set to the direction column. The independent features were the remaining columns This data was encoded and scaled as appropriate. The other columns were passed in as the training data. A test-train split of 30% was chosen after it was found to produce the best accuracies. The hyperparameter, k, was varied from 0 to 100 and the testing and training accuracy plotted for each value of k. The highest accuracy recorded was ∼49%. The test-train split was increased but this resulted in the model becoming overfitted. Overfitted models do not perform well when applied to other data so this situation should be avoided where possible. KNN is a distance-based model, since the dataset included label-encoded data this introduced further away and closer values. Hot encoding is not usually paired with KNN as the increase in the dimensionality of the dataset leads to inconsistencies with the calculated distances (Kirk, 2017).

Linear Regression was next investigated. The ordinary least squares method, outlined in the tutorial notes, and the in-built linear regression on scikit-learn were both used and compared to offer a robust analysis. The code for this section can be found on lines 68 to 79. A subset of the dataset was used as it was found that the entire dataset had a non-linear distribution and was not well suited to linear regression. The model aimed to predict the number of passengers who departed by sea during the summer given the year of travel. The independent and dependent variables were scaled using the min-max scaler. The linear regression model was then split and tested for accuracy. The accuracy was found to be quite poor, with R2 values of train: 0.306 and test: 0.245. Since the training and test scores were relatively similar the model was not thought to be overfitted, however, the two values were monitored closely as a large variance, with the training score higher, indicates overfitting. Lasso and ridge regression were then added to the code to try and refine the model. These methods added the hyperparameter, alpha. Alpha is the coefficient which restricts certain aspects of the model and maintains its generality and its value was experimented with to find the optimal one. For Lasso this was 0.001 and for Ridge was 10. As this model only had one dependent feature the ridge and lasso regressions did not have a big impact on the accuracy. When there are a large number of features this technique is more relevant. The elastic net object was also introduced for robustness but again did not provide very meaningful results due to the simplicity of the chosen dataset. The results of the linear regression model can be seen in Figure 10. This was the worst-performing model, with residuals reaching up to 0.4 for test and training data, as shown in Figure 10 (a).

A Naïve Bayes model was next investigated. It is rooted in probability theory and aims to investigate how the probability of a given feature influences the probability that the target variable will occur. It assigns a probability, the posterior probability, to the event occurring given each of the independent variables has or has not occurred. The test-to-train split was kept at 30-70% to maintain consistency across the models. This model produced similar accuracies compared to the KNN model of about 49%. K -fold cross validation was applied to offer a robust outlook on the accuracy. It divided the data and allowed an average of the randomly allocated test data to be taken.

A support vector machine algorithm was utilised. This is a statistical method of classification that introduces a decision boundary and depending on which side a point lies on the boundary it is classified as one category or another. It allows for a certain degree of misclassification, due to the inherent ‘messiness’ of real-world data, which is defined by the slack variables (Kirk, 2017). In line 104 of the code, the value of C was set. This vector encompassed how the misclassification of the model was managed. The kernel was also defined. This dictated the transformation of the decision boundary. GridSearchCV was also enabled. This refined and found the optimal values of C and gamma.

To compare the KNN, SVM and Naïve Bayes algorithms confusion matrices were plotted. These each showed how many correct classifications the algorithms made concerning whether a passenger was entering or leaving the country. The result of the confusion matrices can be seen in Figure 09.

A diagram of a number of numbers

Description automatically generated with medium confidenceA diagram of a number of numbers

Description automatically generated with medium confidenceA graph of numbers and a number of numbers

Description automatically generated with medium confidence

Figure 09: KNN, Naïve Bayes and SVM Confusion matrices respectively

Figure 08 shows how the three models performed. Interestingly they all had an accuracy of 48-49%.

The right confusion matrix, with the SVM results, shows the heatmap of the optimally chosen poly kernel, slack variable C of 0.1 and gamma of 0.01. The GridCVSearch was implemented and worked by running the various values for all of the variables and returning the ones which were not needed. One reason why this model had a low accuracy may have been due to the relationship between the target and independent features. If the features cannot be split with a certain degree of precision using a hyperplane this method is not suitable (Fletcher, 2008).

None of these modules had an accuracy of over 50%. This could be improved by further refining the SVM decision boundary. There were relatively few features in this dataset so gathering more, such as where the passenger was from, may improve the classification.

A graph with blue dots and red lines

Description automatically generatedA graph showing a line of red line

Description automatically generated with medium confidence

1. (b)

Figure 10: Results of the linear regression models

**Programming**

Programs offer the ability to the user to perform a series of tasks to achieve a particular goal. In this project, the programming language Python was used as it is an accessible high-level language which was well suited to data analysis (Severance, 2009). Many pre-existing libraries were used throughout this project. The packages pandas, seaborn, matplotlib and numpy were used along with the library SciPy, for statistical tools, and sklearn, for machine learning functionality. They have many commonly used functions/methods/objects which streamline the overall process greatly. It reduced the time taken to perform tasks such as filtering, sorting and investigating the data within the data frame (Kirk, 2017). Using these open-source resources also reduces the sources of error as they have been tested and developed over many years. Pandas, for example, began in 2008 and is now one of the most used data analytics tools (“pandas - Python Data Analysis Library,” 2024).

|  |  |
| --- | --- |
| Pandas | .head, .dtypes, .count(), .drop(), sort\_values() |
| Numpy | Arrange, random.normal, zeros, dot, reshape, logical\_not, sum, mean |
| Scipy | Binom, poisson, norm, stats |
| matplotlib | Xlabel, savefig, title, legend, subplots, bar |
| seaborn | Histplot, boxplot, heatmap |
| Sklearn (mostly objects) | KNeighborsClassifier, train\_test\_split, confusion\_matrix, classification\_report, MinMaxScaler, LinearRegression, Ridge |

Table 02: Some frequently used library methods/functions

Table 02 covers some of the frequently used libraries and the methods/functions/objects. Pandas was used to load the originally downloaded CSV file into a data frame. It was then used for much of the EDA. Seaborn and matlibplot were adopted for visualisation.

To make the code run more efficiently and reduce repetition within the code several functions were written. Functions are pieces of code which take input(s) perform a series of operations and then return output(s). Methods have the same functionality as functions but are written within a class. Three functions were written specifically for this project.

The first of the three functions was written to convert the year and month provided in the original CSV file and convert it to a decimal. This needed to be done for the analyses that followed. See line 16 of the Jupiter notebook for the function. The year column was in the form 196101, 196102, 196103, where the 01 was January, 02 was February and so on. The original data type of the year was checked and found to be a float64. This had to be changed to a string before being passed into the function to allow the slicing to function. The function took in a variable, year\_month, indexed up the fourth number and saved this as the year. It indexed to the final two digits, divided them by twelve to get it in decimal form and then returned the two added values. This function was applied to the data frame with the additional column being called decimal\_year, see line 17. As this function was applied to every row of the data frame, 12,934, this function was extremely useful.

Next, a seasons function was put together. This was applied to the same column as above, year. This function definition was outlined on line 22. Again it converted year to a string, and indexed to the second last two digits. It then implemented a conditional statement. Using an if statement it returned the season based on the final two numbers of the string year. This was very useful also as it allowed the rows to be categorised. Again this was applied to every row in the data frame, line 22, which meant it saved a huge amount of time and kept the code as concise as possible.

The final function that was implemented was a function for label encoding, outlined in line 55. This function took in a data frame, made a copy of it and then applied label encoding to each column which had an object data type. It returned the label-encoded data frame.

For loops were also utilised throughout this project, see lines 45, 54, 68 for examples. For loops were particularly helpful during the statistical analysis of this report. A for loop was used to visualise the binomial and Poisson probability mass function for a large number of trials, see lines 41 and 46. The output of these loops was then stored in a list and could be plotted.

While no objects were written directly for this project they were an important aspect of it. One object, in line 68, was taken from the machine learning tutorials. This object was used to perform linear regression using the fit method defined within the object. Pre-defined objects from various libraries were adopted, possibly the most important of which was the data frame object. Each data frame created was an instance of the object which belonged to a class. The class can be thought of as a template with each instance having the same attributes, but different values assigned to these attributes. For example, all data frames have a shape associated with them (“pandas - Python Data Analysis Library,” 2024).

Every attempt was made to make the code as readable as possible. The code was annotated and logical variable names were assigned. Consistent naming patterns were also followed, for example, all data frame variables ended with \_df.

Paradigms are a set of concepts which govern how a computer program is constructed and then run. Most paradigms span across multiple languages which is why it can be helpful to have a grasp on paradigms over languages, as they are more widely applicable (Roy, 2012).

This project borrowed aspects of multiple paradigms. The most linear form of programming is imperative programming. It sees the code as a set of instructions which are to be followed in order. Initially, the project followed this style of programming. In terms of concepts, imperative programming can be thought of as named, deterministic and sequential. This means that it can store values in memory as it runs through, it gives the same result when given the same inputs and it runs one step after another with the next step only starting after the previous has been completed. This had multiple benefits to the project. As many aspects of the program depended on earlier work e.g. the scaling and encoding of the data frame before machine learning, controlling the order it ran was beneficial (Fincher and Robins, 2019; Roy, 2012). Some issues that arose were that as the program developed it became more difficult to locate where various operations were occurring and the code became disorganised.

A procedural approach was then explored. This allowed for a neater approach as it introduced functions. The functions implemented in this project were outlined previously. Functions were called on and applied to the variables that required it. These functions were also suitable to be applied then to other programs. The functions imported and used from the various libraries also fall under the procedural paradigm. The reason these libraries were so popular was that they offered functionality to anyone who implemented them. It allowed the user to tweak them to their application but the main aspects were shared (White and Sivitanides, 2005).

Object-oriented programming (OOP) was another option. This paradigm thinks of a program comprising a series of instances of classes. Objects are based on a class which has a predefined structure and attributes. The big advantage of OOP is that it is better suited to bigger, continuously running programs. There are four key concepts: encapsulation, abstractions, inheritance, and polymorphism (Guerraoui, 1996). Objects were used in this project, for example, KNN was an instance of the class KNeighborsClassifier in line 64. While this was not strictly an OOP it made use of objects. OOP is so popular due to its reusability and adaptability (Guerraoui, 1996).

**Conclusion**

A dataset was obtained, prepared and analysed relating to the passenger movement into and out of Ireland over 49 years. The data was cleaned and prepared. The data was visualised and its characteristics were described using common statistical measures such as its spread. The data was explored in terms of binomial, Poisson and normal probability distributions.

Various ML algorithms were applied to predict if a passenger would be entering or leaving the country. Functions, loops and conditional statements were implemented in a Jupiter notebook.

**Bibliography**

Ananthakrishna, R., Chaudhuri, S., Ganti, V., 2002. Eliminating Fuzzy Duplicates in Data Warehouses.

Ayele, W.Y., 2020. Adapting CRISP-DM for Idea Mining. IJACSA 11. https://doi.org/10.14569/IJACSA.2020.0110603

Behrens, J.T., 1997. Principles and procedures of exploratory data analysis. Psychological Methods 2, 131–160. https://doi.org/10.1037//1082-989x.2.2.131

Bonaccorso, G., 2017. Machine learning algorithms: a reference guide to popular algorithms for data science and machine learning. Packt, Birmingham Mumbai.

Callaghan, N., Tol, R., 2011. UK Tourists, The Great Recession and Irish Tourism Policy.

Chen, C., Härdle, W., Unwin, A., 2008. Handbook of Data Visualization. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-33037-0

Chu, X., Ilyas, I.F., 2016. Qualitative data cleaning. Proc. VLDB Endow. 9, 1605–1608. https://doi.org/10.14778/3007263.3007320

Fincher, S.A., Robins, A.V., 2019. The Cambridge Handbook of Computing Education Research, Google Books. Cambridge University Press.

Fletcher, T., 2008. Support Vector Machines Explained.

Guerraoui, R., 1996. Strategic directions in object-oriented programming. ACM Computing Surveys 28, 691–700. https://doi.org/10.1145/242223.242275

Kaur, P., Stoltzfus, J., Yellapu, V., 2018. Descriptive statistics. Int J Acad Med 4, 60. https://doi.org/10.4103/IJAM.IJAM\_7\_18

Kirk, M., 2017. Thoughtful machine learning with Python : a test-driven approach. O’reilly.

Kumar, A., 2019. Mastering Pandas : a Complete Guide to Pandas, from Installation to Advanced Data Analysis Techniques, 2nd Edition. Packt Publishing, Limited.

Mahon, J., 2021. TRBM1 - Passenger Movement (Number) - data.gov.ie [WWW Document]. data.gov.ie. URL https://data.gov.ie/dataset/trbm1-passenger-movement-number (accessed 1.1.24).

Omar C., W., 2023. Exploratory Data Analysis with Python Pandas: A Complete Guide – Kanaries [WWW Document]. docs.kanaries.net. URL https://docs.kanaries.net/articles/exploratory-data-analysis-python-pandas (accessed 1.1.24).

pandas - Python Data Analysis Library [WWW Document], 2024. . pandas.pydata.org. URL https://pandas.pydata.org/about/

Roy, P., 2012. Programming Paradigms for Dummies: What Every Programmer Should Know.

Schröer, C., Kruse, F., Gómez, J.M., 2021. A Systematic Literature Review on Applying CRISP-DM Process Model. Procedia Computer Science 181, 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Severance, C.R., 2009. Python for Everybody Exploring Data Using Python 3 Charles R. Severance. Charles Severance.

Studer, S., Bui, T.B., Drescher, C., Hanuschkin, A., Winkler, L., Peters, S., Müller, K.-R., 2021. Towards CRISP-ML(Q): A Machine Learning Process Model with Quality Assurance Methodology. MAKE 3, 392–413. https://doi.org/10.3390/make3020020

Tufte, E.R., 2001. The visual display of quantitative information. Graphics Press.

Weiss, N.A., 2012. Introductory Statistics. Addison-Wesley.

White, G., Sivitanides, M., 2005. Cognitive Differences Between Procedural Programming and Object Oriented Programming. Information Technology and Management 6, 333–350. https://doi.org/10.1007/s10799-005-3899-2