A picture containing icon

Description automatically generated

**CCT College Dublin Continuous Assessment**

**Abstract**

*Inbound tourism is a huge part of the Irish economy with as many as 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 in and out of the country and the time of year in which they visited was the aim of this work. EDA was carried out to prepare the data for the statistical analysis and machine learning algorithms that were applied to the dataset to offer insight into the passengers' movement patterns. The information gleaned from these processes was then used to infer what the future trends may be. 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 data resources were explored to find a dataset that contained relevant data regarding the tourism industry in Ireland. After careful consideration, the chosen dataset was downloaded as a CSV file from data.gove.ie (Mahon, 2021). The dataset was then downloaded as a CSV file and uploaded into a pandas data frame using a Python Jupiter Notebook. Multiple Python libraries and packages were used to perform the tasks outlined above. The Jupiter notebook was submitted alongside this report and is referenced where required throughout the report.

# Data preparation and Visualization

You must perform appropriate EDA on your dataset, rationalizing and detailing why you chose the specific methods and what insight you gained. **[0-20]**

To be able to perform meaningful statistical and machine-learning analysis EDA is a critical first step. It 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 patterns that may be present within it. It allows the analyst to become familiar with the context in which the data is set (Behrens, 1997).

The data was found on the data.gov.ie site and had been collected by the CSO (Mahon, 2021). The data dictionary containing the column names and units of measurement associated with the data was viewed in Excel. The Pandas library had very useful tools for performing EDA efficiently (Omar C., 2023). The first 30 rows of the data frame were displayed using the .head() function. The data type of each column was noted. It was decided to rename the column containing the year information in line 6 of the notebook as the original name was not intuitive and this prevented future confusion. It was confirmed that there was no duplicate data within the set. Duplicate data can pose a problem to the integrity of the data set and introduce bias into future analysis (Chu and Ilyas, 2016). It was found that the dataset did not contain any duplicated rows. This is a relevant and observable problem for many datasets, particularly large ones. Research has been done on various ways to manage this problem best while retaining as much of the overall data as possible. Ananthakrishna et al. employed an algorithm which handled missing data in a hierarchical structure in data warehouses with success (Ananthakrishna et al., 2002) .

The shape of the original data frame 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 for ease to remove those rows entirely from the dataset. If the missing data was more significant there were multiple methods which could be used such as imputation and interpolation which could be used (Bonaccorso, 2017; Kumar, 2019).

Next columns with repeated or unnecessary data were removed to keep the data frame as concise as possible. The ascending function was used to identify when the most passenger movement occurred. In August 2008 1,603,381 arrived in Ireland from all airports, the highest recorded value of passenger movement. A function, year\_month\_to\_decimal defined on line 16, was written to convert the year to a graphable quantity. Taking the first entry 196101, which corresponded to January 1961, the function indexed to the final two digits, converted it to a fraction of the year and added the year to the month. A function was also defined to categorise the rows based on season, see line 22. As the tourism industry is seasonal this allowed like to be compared to like in later analyses. Both of these functions were applied to the relevant columns in the data frame.

To gain a better insight into the various subsets of the data the .unique() function was used. This returned all the unique values in a column. 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 using the .isin and .contains functions to select the 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 the passenger numbers here and only the relevant values were filtered for. The .decribe() function was used to ascertain some commonly used statistics and is discussed later in the report.

You must also rationalise, justify, and detail all the methods used to prepare the data for ML (Scaling, Encoding, imputation etc…). **[0-40]**

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.

Taking a look at encoding first, this is the process of assigning a numeric value to categorical data, usually in the form of a string. Scikit-learn library was used to encode the data as it contains many popular methods of encoding (Bonaccorso, 2017). A function was written for label encoding, see line 56. A properly working function for one hot encoding was attempted but in the end this section was done manually on line 56 of the notebook. ~~Label encoding was originally tried, however issues arose during the machine learning analysis. It was considered first due to its relative simplicity.~~ Label encoding assigns ordinal values on nominal data which can be misleading to the ML algorithm which cannot differentiate between the arbitrary values. For instance ‘Arriving’ was assigned a value of 0 and ‘Departing’ a value of 1. To the algorithm ‘Departing’ has a higher value or magnitude which introduces biases and therefore inaccuracies. It also introduced the concept of two or more categories summing to give a third category, see table GG. Logically this makes no sense and for these reasons this method of encoding was not chosen (Bonaccorso, 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 GG: Issue which arose when encoding nominal data

For this reason, one hot encoding was next tried. This created an additional column for each category in the data frame, increasing the computational cost, and assigned a binary value to each, 1 if it contained the categorical variable and 0 if it did not.

Depending on the machine learning algorithm being applied scaling may be required. It involves manipulating numeric data so that the values are brought to closer absolute values but the shape of the original data is retained. For example, regression techniques are heavily influenced by feature scaling as the feature values are inputted and used for further calculation but certain classification techniques such as Decision trees are based on thresholds so scaling is less important Scaling is beneficial when multiple features vary over multiple magnitudes. Although in this case there was only one numeric feature, the number of passengers, since there was a large degree of variance within the feature itself it was deemed worthwhile doing (Bonaccorso, 2017).

The lowest value was 4,815 and the highest was 1,603,381. As it had been established the data was skewed and so min-max scaling was used. The concept behind this was the top value is given a value of 1 and the lowest a value of 0. The remaining values were then assigned a value as a percentage along 0 to 1. This is why it was well suited to this skewed data set as it retained the shape. Standard scaling and L2 scaling were also trialled and the 3 types compared to one another as can be seen in Figure HH. Both min-max and standard scaling retained the data’s shape well. L2 did not work, this may have been due to the large number of outliers within the data set. The decimal\_year values were also scaled so that the values would be of the same magnitude as the passenger number and as they had an even distribution the shape was maintained through the scaling process.

A group of graphs showing different values

Description automatically generated with medium confidence

Figure HH: Comparison of scaling techniques

As previously mentioned the year\_month\_to\_decimal function was employed. This was important for later machine learning

Additionally, a seasons function was adopted. This was decided upon due to the cyclical nature of the tourism industry that aligns with the seasons. This trend could be seen clearly from the scatter plots. Every summer (June, July, August) the numbers arriving and departing peaked. The function took the final number in the Year column and categorised based on it. To be able to compare like with like in machine learning and statistic models made this an important categorisation.

Scaling – values varied by multiple magnitudes, min-max scaling used

Encoded the categorical data (season, method of transport, arriving or leaving)

SMOTE was not used as the classifications had equal numbers of entries.

Appropriate visualizations must be used to engender insight into the dataset and to illustrate your final insights gained in your analysis. **[0-20]**

Visualisation notes

All design and implementation of your visualizations must be justified and detailed in full., referring to Tufts Principles **[0-20]**

Choosing appropriate visualisations is a key decision when conveying data to an audience. It has the power to effectively communicate the information present within the data or it can mislead and confuse. Tuft’s principles were studied and the rationale applied when making the visualisation decisions (Chen et al., 2008; Tufte, 2001).

**Statistics**

You need to analyse the chosen dataset using statistical logic and statistical techniques. Note: ALL Statistical work MUST be carried out using Python.

You are required to

**Summarise your dataset clearly, using relevant descriptive statistics and appropriate plots. These should be carefully motivated and justified, and clearly presented. You should critically analyse your findings, in addition to including the necessary Python code, output and plots in the report. You are required to plot at least three graphs. [0-35]**

It is critical to be able to clearly and concisely describe the characteristics of the dataset being analysed. It provides an insight into the overall information contained within the dataset, the types of variables and the relationship between them. It is the first stepping stone within the overall analysis and often guides the subsequent steps (Kaur et al., 2018).

The Python pandas library was utilised to maximise efficiency and accuracy. The .types function was used to determine the data types that were contained in the data frame. The number of passengers, with column name VALUE, was numeric, as was the derived 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. Half of a person cannot travel and therefore this variable consisted of only whole numbers. The value could also be classified as a ratio variable as it has a definite zero point.

The dataset was obtained from data.gov.ie and was collected by the CSO. It is representative of the population who travelled from 1961 to 2010.

The measures of central tendency were taken for the dataset next. The .describe pandas function was used. 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 graph with a number of passengers

Description automatically generated

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

This indicated that the data was heavily skewed and this was visualised by plotting the values on a histogram. In Figure AA the data is observed as right skewed with the binwidth unspecified. 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).

An initial box plot was created containing all the data which showed there were a lot of outliers. For this reason, 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 BB 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

IQR

Lower Quartile

Figure BB: Comparison of Boxplots

The data frame was filtered to show data pertaining to 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 are 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 BB. The boxplot illustrates the variability in air compared to sea travel. While sea travel has remained relatively steady the aviation sector has increased significantly from 1961 to 2010. The IQR of the yellow plot is 40,590 and the blue is 117,161. This measure of spread indicates that there is much more variability within the air than sea travel sector. (add 5 number summary?)

A graph with many colored dots

Description automatically generated

Figure CC: Scatter plot from 1961 to 2010

The third chosen plot was a scatter plot, shown in figure CC. The year is plotted along the x-axis and the number of passengers moving in millions is along the y-axis. Up until the 1990s, the data remains 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, both incoming and exiting the country. This reflects what was observed in the boxplot with the large variation in data for air travel. There is consistently more incoming passengers than departing for air travel, but the opposite is true for sea travel. Scatterplots are invaluable for showing the relationship between two variables and in the case of time data it can show trends over the given period. Air travel is clearly increasing while sea travel has remained relatively stable over the 50 years presented.

Use two discrete distributions (Binomial and/or Poisson) to explain/identify some information about your dataset. You must explain your reasoning and the techniques you have used. Visualise your data and explain what happens with the large samples in these cases. You must work with Python and your mathematical reasoning must be documented in your report. [0-30]

Probability distributions allow a numeric value to be assigned that an event will occur. Depending on the circumstances in question different distributions are appropriate. 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 conditions, and compare the chances that various situations will arise based on collected data (Weiss, 2012).

One such application of probability theory is the binomial distribution. It is a series of events or trials where either an event occurs or does not. A probability value Is assigned to both, the sum of which must be added to give 1. The events must be independent of one another, i.e. the probability of a passenger entering Ireland does not affect the probability that the next person is entering. Due to the large number of people who entered and left over the extensive 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. As previously discussed and shown in figure CC, sea travel was relatively constant over the 50-year time frame both for departing and arriving. This key feature meant that it was more suitable for a binomial distribution than the entire dataset.

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 event which was chosen 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. It was checked that the two probabilities summed to give 1. The binom class within the scipy.stats library was used to calculate the probabilities.

Various questions were investigated. Taking an initially small number of trials, 10, and experimenting with the scipy.stats functionality a range of results were found.

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 the 10 and the y-axis the probability that the given person was departing from Ireland. Due to p ≈ q the distributions are symmetric. Of the 10 trials, the probability that 5 of the 10 were departing is at its highest value, calculated by the pmf method. The .cdf method calculates the probability of k or less successes and .sf finds the probability of k or more successes. The results of these can be seen on figure DD. The mirror effect can be seen here. This would be mirrored in n of any size with the heights of the bars changing.

A graph of different colored lines

Description automatically generated

Figure DD: 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. This was a simple calculation and over the long term, 4.99 ≈ 5 in every 10 randomly chosen sea travellers would be departing from the country.

**A graph with a line

Description automatically generated**

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

Finally, the probability for a large number of trials half of which resulted in a success was plotted. This shows that for a large n value 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 binomial distributions with large n values than normal distributions (Weiss, 2012).

Next, a Poisson distribution was explored. The total number of people moving by sea was calculated and divided by the total number of months (one row corresponded to one month). 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. 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 line

Description automatically generated

Figure ZZ: Poisson distribution over a long time frame

Use Normal distribution to explain or identify some information about your dataset. [0-20]

A supplementary categorical value was added to the dataset, season, to create a subset of the data which was normally distributed. Figure FF 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 normal distribution

Description automatically generated

Figure FF: 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. In practical terms, 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%. This information extracted from the data was useful as if this travel remained normal

**Explain the importance of the distributions used in point 3 and 4 in your analysis. Justify the choice of the variables and explain if the variables used for the discrete distributions could be used as normal distribution in this case. [0-15]**

Binomial, Poisson and Normal distributions offered a way of assigning values to probabilistic events in terms of passengers entering and leaving the Ireland.

As visualised in figures EE and ZZ the probability mass function tended to the limit of zero as the number of trials increased and the time interval increased.

# Machine learning for Data Analytics

Over and underfitting!!

Explain which project management framework (CRISP-DM, KDD or SEMMA) is required for a data science project. Discuss and justify with real-life scenarios. Provide an explanation of why you chose a supervised, unsupervised, or semi-supervised machine learning technique for the dataset you used for ML modeling. **[0 - 20]**

CRISP-DM:

Cross-Industry-Standard-Process model for Data Mining (CRISP-DM) is a widely used data management framework in industry. Many data projects do not follow a specific framework but doing so can reduce costs and completion timelines, as well as 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). For this reason is has been described as industry independent. This indicates it has a high degree of flexibility, an advantageous attribute (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 of CRISP-DM were discussed by Studer et al. CRISP-DM is focused more on static analysis and is not set up to handle real-time data. This limits the dynamic decision making capabilities of this approach in a machine-learning setting. 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 models have a wide range of uses, including prediction, classification, and clustering. It is advised that you assess several approaches (at least two), choose appropriate hyperparameters for the optimal outcomes of Machine Learning models using an approach of hyperparameter tunning, such as GridSearchCV or RandomizedSearchCV. **[0 - 30]**

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, i.e. was the passenger entering or leaving the country. The independent features were the remaining columns Type of Passenger Movement, number of passengers, decimal\_year and Season. 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. 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%. This was a low accuracy. 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. The confusion matrix showed that 749 of the arriving class and 853 of the departing class were correctly classified. 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 to reduce the influence of the varying magnitudes of in the passenger data. The linear regression model was then split and tested for accuracy using the sklearn.linear\_model linear regression method. 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. 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.

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. For example, if the probability of a passenger arriving is larger given that it is summer then the model would return arriving when the independent variable was summer. Again the sklearn library was used and the Gaussian distribution object was imported from it, see line 99. 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 divides the data and allows an average of the randomly allocated test data to be taken.

Finally, 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. 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. It was attempted to include the kernel type as well however the computational time was too long so this ideal value could not be found.

Show the results of two or more ML modelling comparisons in a table or graph format. Review and critically examine the machine learning models' performance based on the selected metric for supervised, unsupervised, and semi-supervised approaches. **[0 - 30]**

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. Confusion matrices offer an effective and clear way to compare how multiple techniques performed in classification problems. The three confusion matrices were plotted with the same y-axis scale so that a fair comparison could be drawn. The result of the confusion matrices can be seen in Figure GG.

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 GG: KNN, Naïve Bayes and SVM Confusion matrices respectively

Ideally, the two diagonal boxes would contain the correctly classified categories. Figure GG 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).

The Naïve Bayes

None of these modules had an accuracy of over 50%. Th

A graph of a graph showing a line graph and a diagram

Description automatically generated with medium confidenceA graph with blue dots and red line

Description automatically generated

Figure FF: Results of the linear regression models

Demonstrate the similarities and differences between your Machine Learning modelling results using the tables or visualizations. Provide a report along with an explanation and interpretation of the relevance and effectiveness of your findings. **[0 - 20]**

Linear regression – lasso, ridge, elaticnet

KNN, Support vector, naïve bayes

Over and under fitting

**Programming**

The project must be explored programmatically, this means that you must implement suitable Python tools (code and/or libraries) to complete the analysis required. All of this is to be implemented in a Jupyter Notebook. Your codebook should be properly annotated. The project documentation must include sound justifications and explanations of your code choices (code quality standards should also be applied). **[0-50]**

|  |  |
| --- | --- |
| 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 |

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. These libraries and packages are incredibly useful. They have many commonly used functions/methods 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).

Table GG: Some frequently used library methods/functions

Table GG covers some of the frequently used libraries and the methods/functions. Pandas was used to load the originally downloaded CSV file into a data frame. It was then used for much of the EDA. It provided very useful tools to gain insight into the columns, rows, the commonly used descriptive statistics and so on. Numpy was brought in to perform mathematical operations and matrix operations. Scipy and sklearn were used for statistics and machine learning reqpectively. 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. As previously mentioned many functions exist and are available through open source resources online. Three functions were written specifically for this project. A fourth function was tried but it was not working as required so the code was written manually.

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. It saved time and converted the year to a form that could be worked with.

Next, a seasons function was put together. This was applied to the same column as above, year. This function definition was outlined on on line 22. Again it converts year to a string, and indexes to the second last two digits. It then implements a conditional statement. Using an if statement it returns 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. Tourism is a seasonal industry and labelling a row as winter, summer, spring or autumn was helpful in subsequent steps. 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. It was also attempted to write a function for one hot encoding, however, the loop was not working as expected and was producing extra rows after each iteration. It was decided for ease to do this manually instead.

For loops were also utilised throughout this project, see lines 45, 54, 68 for examples. Again these helped to improve the overall efficiency of the code and kept it concise. For loops were particularly helpful during the statistical analysis of this report. For loops are designed to iterate through a fixed number of loops, unlike while loops which can run indefinitely. 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 belongs 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. The original passenger\_movement\_df had a shape of 12936 rows and 10 columns (“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 variable ended with \_df. No values were ’hard coded’, variables were used so that changes to the early parts of the code did not propagate down the code.

In a dedicated section in your report, discuss your use of aspects of various programming paradigms in the development of your project. For example, this may include (but is not limited to) how they influenced your design decisions or how they helped you solve problems. Note that marks may not be awarded if the discussion does not involve your specific project. **[0-50]**

Paradigms are a set of concepts which govern how a computer program is constructed and then run. While there might be one ‘correct’ paradigm for every application, some may be more suitable than other depending on the situation. Most paradigms span across multiple languages which is why it can be helpful to have a grasp on paradigms over languages, 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 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 in 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.

For this reason a procedural approach was then explored. This allowed for a neater approach as it introduced functions but maintained the ordered line by line approach. The functions implemented in this project were outline 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. It increased the reusability of the program. The functions which were imported and used from the various libraries also fall under the procedural paradigm. The reason these libraries were so popular was that they offer functionality to anyone who implements them. It allowed the user to tweak them to their own application but the main aspects were shared (White and Sivitanides, 2005).

Object oriented programming (OOP) was another option. This paradigm thinks of a program being comprised of a series of instances of classes. Objects are based on a class which has a predefine structure and attributes. Objects run in the same state that perform operations. The big advantage of OOP is that it is better suited to bigger, continuously running programs. There are four key concepts: encapsulation, abstractions, inheritance, polymorphism (Guerraoui, 1996). Objects were used in this project, for example knn was an instance of the class KNeighborsClassifier in line 64. By pressing shift tab the attributes, such as n\_neighbours, can be seen. Another example of an instance of an object can be seen in line 55, lencoder. Making use of open source objects was very useful as it reduced the complexity of the code greatly. 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**

**Bibliography**