Transport mode preference by students and staff

Year: 2021-2022

In partial fulfilment of the requirement of the degree Master of Science in Data Analytics

By
Thanuja Modiboina
Student Number:210274503

Under the supervision of
Dr.Sotos C Generalis
Head of Mathematics Department
College of Engineering and Physical Sciences
Aston University

Abstract

Aston university supports sustainable travel and net zero emissions in Birmingham city. To achieve these, the university must know about the travel behaviour of students and staff. So, the university conducted a commuter survey to know students and staff, how they currently travel to and from the university, the important issues surrounding their journey, and what changes they might expect over the next few years. What is the distribution of percentages of travel modes to/from the university and Which features affect travel modes? This study comes under the Multi-output classification problem, to analyze these we used a Multi-output classifier ensemble with a Random Forest classifier and Gradient boosting classifier models. The result of these models is the demographic features such as whether they are students or staff, gender, the distance from house to the workplace, age group, mode change between summer and winter, reasons for why they are choosing those travel modes, number of trips per week to the university and time taken to travel the university. From the percentages of travel modes to/from the university, most of the staff are using the car. The accuracies obtained by these models are in the range of 80% and 95%.

Contents

1 Introduction **3**

2 Context/Literature Review **4**

3 Problem description **6**

4 Methods **7**

4.1 Data collection 7
4.2 Data cleaning 7
4.3 Exploratory Data Analysis 8
4.3.1 Descriptive statistical analysis 8
4.3.2 Descriptive graphical analysis 9
4.4 Data preparation 14
4.4.1 Dealing with missing values 14
4.4.2 Feature selection 15
4.5 Machine learning models 16
4.5.1 Multioutput classifier 16
4.5.2 Decision Tree[18] 17
4.5.3 Ensemble methods[6] 19
4.5.4 Random Forest classifier[16] 19
4.5.5 Gradient Boosting classifier[13][15] 20

5 Evaluation **24**

6 Conclusion **26**

7 Appendices **29**

1 Introduction

A survey reveals all reasons behind every problem. If we conduct a survey, few people will respond, and few will intentionally or not, give incorrect answers. There are few commuter surveys held in some places to know about their employees or people. The transport mode choice is the main interesting topic in every city. If traffic congestion is high, then the emission of gases is high. So, the traffic demand and emission of gases are directly proportional to each other. We must focus on which transport releases more emissions of gases into the environment. Based on that we must increase transport modes which produce less emission of gases. For example, cars are producing more emissions of gases instead of using cars it's better to choose which vehicle produces fewer emissions. If we reduce these gases, the air quality and climate change will be improved. There are so many studies done regarding transport mode preference in post-pandemic. Transport should not be a barrier to those who are accessing education, work, services, and facilities. Stated preference surveys (SPS) are crucial techniques that help in predicting choices by asking respondents about their potential options in hypothetical circumstances given a certain set of circumstances [14].

Here, our main aim is to predict based on which aspects people are coming to/from Aston university by travel modes. And what are the percentages of travel modes i.e., the percentage of staff and students that come by car, bus, train, metro, cycle to work, e-scooter, by foot etc. and how people get on campus? And what is the distribution of transport modes for students and staff and what barriers do they have regarding transport? The university will understand the priority of their travel mode from these results. So, based on that, we will try to reduce using cars and make our campus a car-free campus. Our big goal is to make the campus car-free. For this, we must enhance our campus with on-campus signage, build new paths, improve transit routes, and parking and transportation services work together and it will give the option to leave their cars by students. Reducing this carbon footprint includes designing features to encourage bicycling, walking, carpooling, and the use of electric vehicles [17].

Working to improve these connections will be crucial for the University in terms of facilitating mode shift as well as because the quality of the city is key for attracting and retaining staff and students, which will enable the University to expand. Making this will reduce emission gases and it will help to UK's government('The Climate Change Act'). There are so many advantages like it will reduce traffic, and pollution and make a healthy environment. Climate change is the biggest challenge in the UK and the world. According to the research, global temperatures are increasing, causing changes in the weather, the level of the sea is increasing, and an increase in the frequency and severity of extreme weather events. Due to this global warming, sudden changes happen in the environment like floods, temperature variations, rainfall, cyclones etc. To avoid these, we must concern with travel behaviour. According to reduce these, try to use Eco-friendly travel modes.

The Birmingham City Council and Transport for West Midlands are making significant investments to improve the area for biking and walking. A city that will allow all age groups of people can live with happiness and be healthy. They can take clean air and provide opportunities to enjoy cycling and walking. There are so many impacts of air pollution and noise pollution, which makes health severe. The lack of exercise leads to obesity, heart disease and cholesterol. The number of adults in England who are obese, or overweight has increased (28% are obese and 36% are overweight) and childhood obesity also increased. To handle these health diseases, we must choose walking and cycling. This will improve the quality of air, and those living near the city will be healthy. Providing discounts and further improvements regarding travel modes can reduce using cars. From these opinions, we can solve issues on campus regarding parking and commuting requirements. According to the study, when the temperature is low, people prefer instead of

biking they will come by car or bus, whereas in high temperatures they come by walk or bike [3].

The transport mode preference is based on the distance from the house to the workplace, travel time, weather conditions, security reasons, income, salary etc. This project will give a clear idea regarding transport modes. Most of the papers done after the pandemic those papers representing those people coming by public transport decreased. So, they conducted surveys and did this analysis to know the reasons behind this rapid economic loss in public transport in every city. Due to Covid-19 so many people preferred by coming in private cars and refused to come by public transport and shared minibuses [4]. This study aims to assess the factors influencing commuters' decision to use private vehicles instead of public transportation, as well as their perceptions of the risks involved, with a focus on potential implementation strategies that would ensure the long-term sustainability and resilience of our public transit system [8]. This transportation strategy identifies activities for improving public domain and transportation outside of the university while supporting the university's design and plan implementation. Others must take these steps to promote the growth of Aston University and the larger Knowledge Area and provide them with the chance to fully participate in the continuing transformation of Birmingham and the surrounding area.

The change in travel mode shifts and increase in greenhouse gas emissions during the pandemic year 2020. they are pre-pandemic, post-pandemic and mid-pandemic. Travel to campus was significantly reduced as a result of the epidemic. At the beginning of the outbreak, the proportion of people who continued to commute to campus by automobile roughly doubled. The biggest mode shift seen at the start of the pandemic was the switch from walking, bicycling, and taking public transportation to drive.[10] These problems occurred during the pandemic everywhere. Everyone is afraid to travel on public transport, rail and bicycles due to this Covid.

University students' commute habits have come under more attention in recent years. The reason for this interest is mostly because students are a key target group in the behaviour change process [5]. A greater number of students travel from house to university and university to home. Most of the carbon footprint is contributed by students' travel trips to the university.

2 Context/Literature Review

The US government, and federal are trying to spend money to expand public transportation in urban areas. The research aims to attract people out of their cars. This study examines the specific attributes of choice passengers and the environmental factors (such as transit service quality, driving conditions, built-up areas, etc.) that influence their decision to use public transportation. People might not immediately leave their automobiles when traffic congestion increases. This research finds features of multi-modal transportation where transit is successful in attracting discretionary users by adjusting for elements that can restrict access of a car-owning individual to the household vehicle [7]. Here, two analyses were used to get the results. First, descriptive analysis is to investigate the situations in which commuters from families with cars use transportation. And then multinomial logistic regression model is used to determine how elective transport mode choice between commuters who own cars is affected by relative transport service quality [7].

Impact of Covid-19, most people used private transport mode instead of public transport mode. The online questionnaire survey was held in India. Based on that survey, they used a logistic regression model to predict which factors are affecting the modal shift. This model indicates that the commuter's demographic features such as gender, location, age, and monthly income seem to impact travel mode preferences. And there

is a correlation between trip parameters such as travel cost, time taken to reach the destination, cleanliness and crowd associated with this mode shift from public transport to using the car. In summary, efforts must be focused on regaining the consumers' faith and trust by giving them access to a secure, safe, and clean environment for those who are using public transport. As a result of the findings, journey duration is positively correlated with the mode-switching switching behaviour of respondents. In essence, when the trip time on public transportation increases, respondents are more willing to convert to private automobile commuting. [8].

This analysis is about the impacts of mode choice based on the geographical and annual weather variations using weather and travel survey data in Sweden. There was a discussion about the connections between weather indicators and mode distributions. Here, they used multinomial logit models in research to examine how weather influences mode choice in Sweden. They discovered that people are more likely to choose to walk and take public transportation during the cold and are less likely to choose to cycle. In contrast, compared to other alternative modes of transportation, the probability of opting to travel by private automobile falls under warmer weather conditions. This article concludes by comparing the models' outputs for different locations and seasons, and by diverse behaviours resulting from seasonal and regional variations on weather impacts in Sweden (or the study also demonstrates that traveller's impressions of the weather vary depending on where they are and the season they are travelling in, which has a significant impact on their travel behaviour)[12].

This research investigates how passengers use intercity buses and trains as alternate modes of transportation for the LDT(Long distance travel). A questionnaire survey was conducted in Bangladesh's capital city of Dhaka for all age group people. From this survey, demographics like age, gender, location etc. and many aspects of mode choice are collected. Here, four models(Naïve Bayes, Decision tree, Support vector machine, Random Forest, and K-Nearest neighbour) are used for transport mode preference. But only Random Forest(RF) gave 95% accuracy which is the best-performing model among these models. The feature importance is obtained from the model. The most important factors in predicting the choice of travel mode are the taken for travel stops or stations closer to the destination, route location, and the number of times of travel (frequency)[1].

According to the research's results, the elements that influence the mode of transportation that employed persons in Gaza City choose to include the total journey time, total travel cost split by personal income, ownership of transport means, age, distance, and average family monthly income. Here, the logit model is used to get results. This logit model is a simple and popular discrete choice model when compared to all other models. Two features are not important at a 90% of confidence level. They are gender and out-of-vehicle time, which are removed from the model. Because the proposed model is accurate to a 95% level of confidence, it can forecast the behaviour of employed persons in Gaza City. Out of this investigation, some recommendations have been made. They are this model can be used to analyze travel demand and develop transportation plans in Gaza city. Encourage young people to use bicycles and the mode choice should be modelled using the data from the developed model [2].

Predicting travel behaviour, which is the most modelled transit choice, is one of the most crucial components of transportation modelling. It involves elements of human behaviour that are concerned with making choices. A disadvantage of aggregate models is that they can only anticipate and estimate travel choices using collected area data. Aggregate models are often employed to deal with the travel choice behaviour of individual travellers. Here, an approach is used to examine and forecast traveller choices which is called a discrete choice analysis. This model is a mathematical function which calculates the likelihood that a person would choose a certain route based on the maximization of utilities. The discrete choice model was developed using Stated survey data, which

includes data collection on personal decisions, personal attributes, and different alternatives for trips. The logit model is a popular model among them for validating the mode selection due to its ease of building the model and greater accuracy when compared to the other types [11].

In this paper, we have studied how university students' commuter habits have altered in response to Portugal's most recent financial crisis. Especially for longer distances, the recession created a significant incentive to reduce the expenses of the most expensive transportation options. Automobile users appear to prioritize comfort and leisure over money, but our findings imply that this may not hold over longer distances, where money becomes a more important factor. This discovery could provide an opportunity to adopt some innovative policies at greater ranges. This should provide us with the opportunity to consider how we may, in line with the general trend in other parts of Europe, raise the cost difference between driving a car and taking public transportation inside metropolitan areas. They have also confirmed that the variables that support the usage of a certain mode of transportation at one time may alter in another period based on the model's results. Here, the Decision tree model is used to get the results. The change in explanatory factors over time is significant because it requires policymakers to take their dynamics into account when building analytical and decision-making models. Our research also highlights the significance of considering key macrosocial contextual elements, including sociocultural, political, and economic aspects [5].

3 Problem description

Aston University has undertaken travel surveys of its students and staff for several years, this survey was held in November 2019. It is conducted one commuter survey. This survey is to know about issues behind student and staff journeys. In the upcoming years, Aston university will develop with subsequent additional users and commuting requirements. This survey mainly focuses on how students and staff travel to/ from Aston university. From this commuting survey, we will get opinions about travel from students and staff. From these responses, Aston University's transport strategy will take decisions regarding travel mode and make it helpful to students and staff. The strategy also addresses several issues that the university has recognized as important, such as: ensuring that getting to the university won't be a barrier to getting an education or a job, providing parking space on campus, improving the public environment and the experience of being on campus and for workers, students, and tourists, the transition to sustainable transportation should continue.

Aston university is located in the city centre, the train terminals and bus stations are closer to the university campus. However, the walking and bicycling environments are sometimes not good. And there is a large amount of severance due to the motorways between the university and the rest of the city centre. These issues are not solved by the university.

The transport strategy also acknowledges and addresses several additional 'drivers for change' that make business as usual in the transportation industry unsustainable. These factors indicate that it is urgently necessary to decrease the use of cars and promote the use of active, sustainable, shared, and efficient means of transportation. This strategy tries to enhance public transportation and improve connections to the city centre.

From the survey, Aston university received responses from both students and staff. Based on this data we are predicting the below requirements. In this project, what is the distribution of transport modes for students and staff, what are the percentages of how people get into the campus by bus, rail, metro, walk, cycle, car share, car passenger dropped off, other etc. and why would people be choosing the way they travel to university and what would encourage them to change the travel modes? Based on which features

they are choosing the way they travel to/from the university(which features are affecting them to choose those travel modes).

4 Methods

4.1 Data collection

Dataset1 link: dataset1
Dataset2 link: dataset2
These files contain students' and staff's responses from the survey data.

This data is collected from Aston university's students and staff. Those who responded to this commuters survey gave their response which consists of their travel opinions. We have commuters survey data from students and staff separately. Both data files are in the .xls type. First, we must read these files in python. The file 'Sheet1.xls' (df1) consists of student data and 'Sheet2.xls' (df2) consists of staff data. To further process, we merge these two files into one dataset(df3). The present dataset consists of 755 rows of observations and 161 columns or features which means 755 people responded to this survey. In every dataset first, we must clean the dataset and transform the data into a certain format then be ready to apply algorithms and deploy models.

There are a few steps to clean the dataset. They are Check the structure or shape of the dataset i.e., how many observations and features are in the dataset. Handle missing values in all features. Identify the features which are not useful to predict travel behaviour. Mention the type of all features in the dataset, which are numerical or categorical. Check the type if we want to change the type to do further process. Then encode the data if it contains categorical features.

To know the basic information, we use info() and describe(). The info() function gives the shape of the dataset and data types of features. 123 features are float type and 38 features are object type in or dataset. Those features are referred to as numerical if it has an int or floats data type and categorical if it has an object datatype or string values. The describe() function which gives descriptive statistics of both numeric and categorical features.

Here, we are removing unwanted columns which not useful for our prediction. (1. Do you consent to your data being processed as described above? , 3.a. If you selected Other, please specify, 7. How do you usually travel TO Aston University? 34. If you would like to be entered into the prize draw for your chance to win £50 shopping vouchers, please provide your Aston e-mail address.). The column names are too large, so we renamed them.

4.2 Data cleaning

In the given data, we must check missing values, duplicate values, outliers, and irrelevant features that are not useful in prediction. Our dataset has no duplicate values. But we have null values in a few columns. So, we must handle missing values.

First, we used the fillna() method to fill in the missing values of a few columns. These missing values are filled with 0's in those columns('7.1.a. Bus – Monday', '7.1.b. Bus - Tuesday', '8.1.a. Bus - Monday','8.1.b. Bus - Tuesday', etc. nearly 120 columns are filled with 0's).

Now, we have 157 columns in our dataset. This dataset consists of which day people are coming by bus, rail, metro, car-share, walk etc. we can fill a few columns('7.5.f. Bus – Every day',' 7.2.f. Rail – Every day', '7.3.f. Metro – Every day', etc.) by summing multiple columns('7.1.a. Bus – Monday','7.1.b. Bus – Tuesday','7.1.c. Bus – Wednesday','7.1.d. Bus – Thursday','7.1.e. Bus – Friday','7.5.f. Bus – Every day', etc.). Then remove

those columns such as '7.1.a. Bus – Monday', and'7.1.b. Bus – Tuesday','7.1.c. Bus – Wednesday','7.1.d. Bus – Thursday','7.1.e. Bus – Friday', etc. except '7.5.f. Bus – Every day','7.2.f. Rail – Every day', etc. After removing these columns, only 55 columns in our dataset.

After handling those columns and then find the percentage of missing values in every column. If the percentage of missing values is >80%, then remove those columns. ('8.a. Other', '10.a. If other please explain in more detail., '11.if other', '12.a. If other please explain in more detail.', '18.a. If you selected Other, please specify:', '27.feedback on the cycle parking available on-site at the University', '33.comment on or suggest improvements to the general campus commuting infrastructure'). These columns are having >80% of missing values.

The feature '4.postcode' is not required for our analysis. Because already we have the features which represent how much time it will take from their home and the distance from home to the workplace or university.

The feature '16. At what time do you normally depart from campus?' which contains missing values filled using value_counts(). This value_counts() function will give those columns that have the most frequent value(count of unique values). The order of those values is decreasing order, so the first element is the most frequent value. The missing values of this feature are filled with the most frequent value. here why we used this value_counts() method is this feature contains only 8 missing values. This method used the mode or most occurring value to fill in missing values.

Now we have the data about how many days students and staff are travelling to/from the university bus, rail, metro, walk, cycle, drive all the way, car share, park and ride (E.g., car to train/bus then public transport), other etc. From this data, we are calculating the percentage of students and staff getting into the campus by various travel modes. Then sum each column separately based on member features such as '7.5.f. Bus - Every day', '7.2.f. Rail - Every day', '7.3.f. Metro - Every day', etc. Sum all 10 columns. (Percentage means the number of items divided by the total number of items). In our case, the sum of each vehicle column is divided by the sum of all vehicle columns based on staff and students' travel to the university. Then sum each column separately based on member features such as '8.1.f. Bus - Everyday', '8.2.f. Rail - Everyday', '8.3.f. Metro -every day" etc and repeat the same procedure for travelling from university to home.

Then percentages of students and staff travel to/from the university by various travel modes are calculated.

4.3 Exploratory Data Analysis

EDA is an important step in Machine learning. To create a valuable product using the data, you need to explore the data. Explore and analyze the data, there are two methods in exploratory data analysis. They are descriptive statistical analysis and descriptive graphical analysis.

4.3.1 Descriptive statistical analysis

Descriptive statistical analysis, which gives information about statistical parameters of numerical features.

```
df3.describe()
```

	7.5.f. Bus - Everyday	7.2.f. Rail - Everyday	7.3.f. Metro - Everyday	7.4.f. Walk - Everyday	7.5.f. Cycle - Everyday	7.6.f. Drive all the way (lone driver) - Everyday	7.7.f. Car passenger (dropped off) - Everyday	7.8.f. Car share (with someone at the University) - Everyday	7.9.f. Park and Ride (E.g. car to train/bus then public transport) - Everyday	7.50.f. Other (please specify) - Everyday	8.1.f. Bus - Everyday	8.2.f. Rail - Everyday	8.3.f. I - Ever
count	755.000000	755.000000	755.000000	755.000000	755.000000	755.000000	755.000000	755.000000	755.000000	755.000000	755.000000	755.000000	755.0(
mean	1.221192	1.231788	0.131126	1.124503	0.315232	0.777483	0.147020	0.120530	0.292715	0.092715	0.674172	0.823841	0.0(
std	2.126023	2.106714	0.791214	2.064702	1.155643	1.733818	0.791769	0.717378	1.151812	0.584401	1.535541	1.766762	0.5
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0(
25%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0(
50%	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.0(
75%	1.500000	2.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.0(
max	10.000000	9.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	7.000000	5.000000	6.000000	6.000000	6.0(

Figure 1: Descriptive statistics of the dataset

Descriptive statistical analysis is used to know about statistical parameters like mean, median, count, minimum value, maximum value and standard deviation etc. These are the parameters which explain numerical features. For categorical features, this gives parameters like count, frequency of value, top value and unique value etc.

4.3.2 Descriptive graphical analysis

Descriptive graphical analysis is analyzing the data with the help of graphs. Already we calculated percentages of students and staff travel from/to the university by travel modes. Based on those percentages here plots are plotted. Here, the percentage is calculated as the sum of the values of the column divided by the total sum of all columns multiply by 100.

percentages of students that come by travel mode to the university:

This plot shows the distribution of transport modes by students travelling to the University. From this plot we can clearly say, students, are choosing mostly to walk, bus and rail.

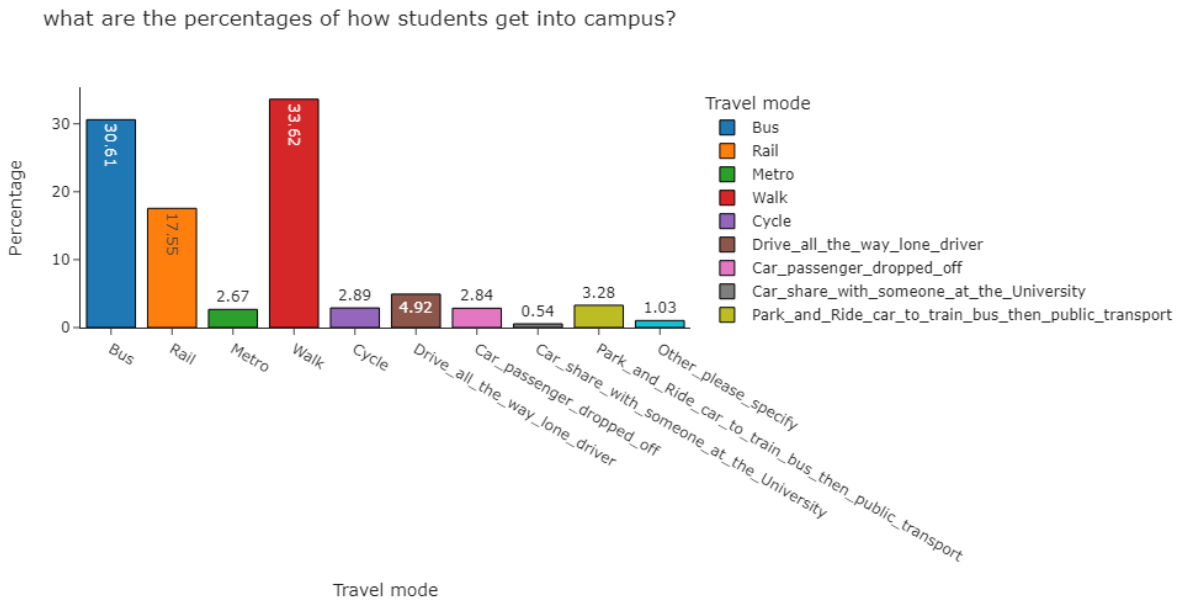

Figure 2: percentages of students travel to the university by various modes

percentages of staff that come by travel mode to the university:

This plot shows the distribution of transport modes by staff who travel to the University. From this plot, we can observe that the staff are choosing mostly to rail, drive all the way (lone driver) and bus. But this is not good to encourage. Because most of the staff are coming by car alone.

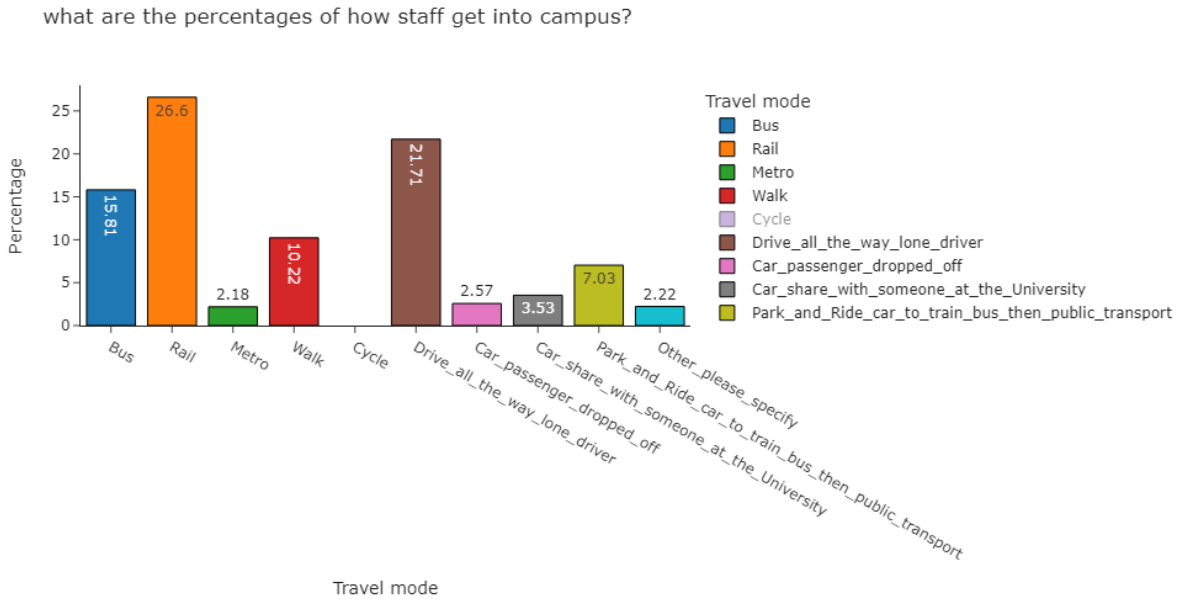

Figure 3: percentages of staff travel to the university by various modes

percentages of students that come by travel mode from the university:

This plot shows the distribution of transport modes by students who travel from the University to home after work. From this plot, we can observe that the students are choosing mostly to walk, bus and rail. But 5.39% of students are coming by driving all the way alone.

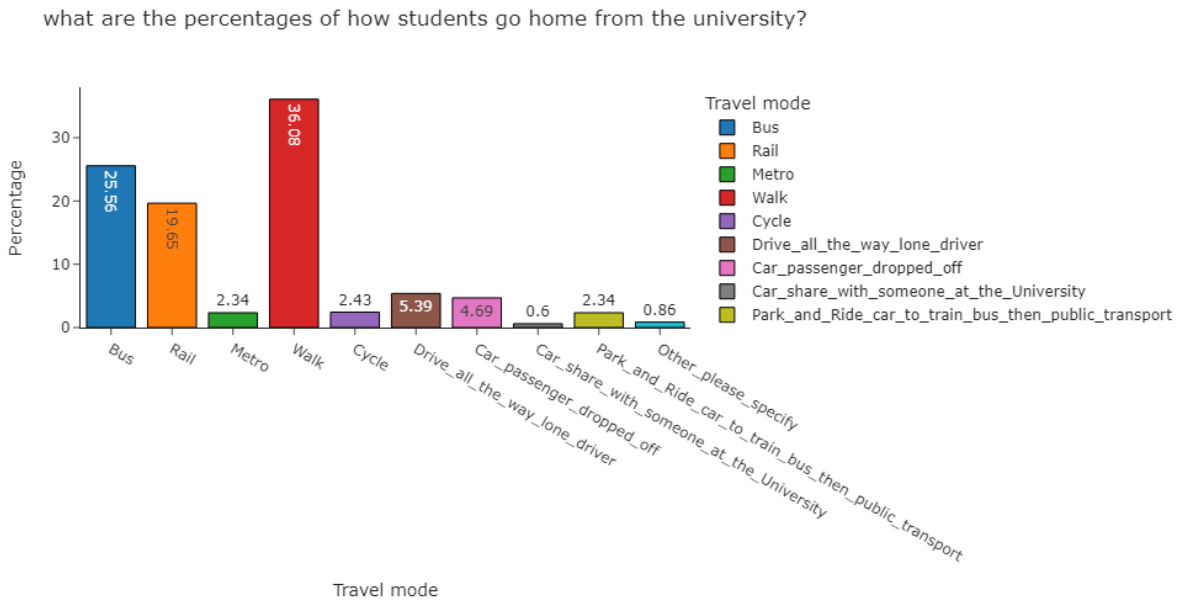

Figure 4: percentages of students travel from the university by various modes

percentages of staff that come by travel mode from the university:

This plot shows the distribution of transport modes by staff who travel from the University to home after work. From this plot, we can observe that the students are choosing mostly rail, driving all the way alone and bus. But 5.39% of students are coming by driving all the way alone.

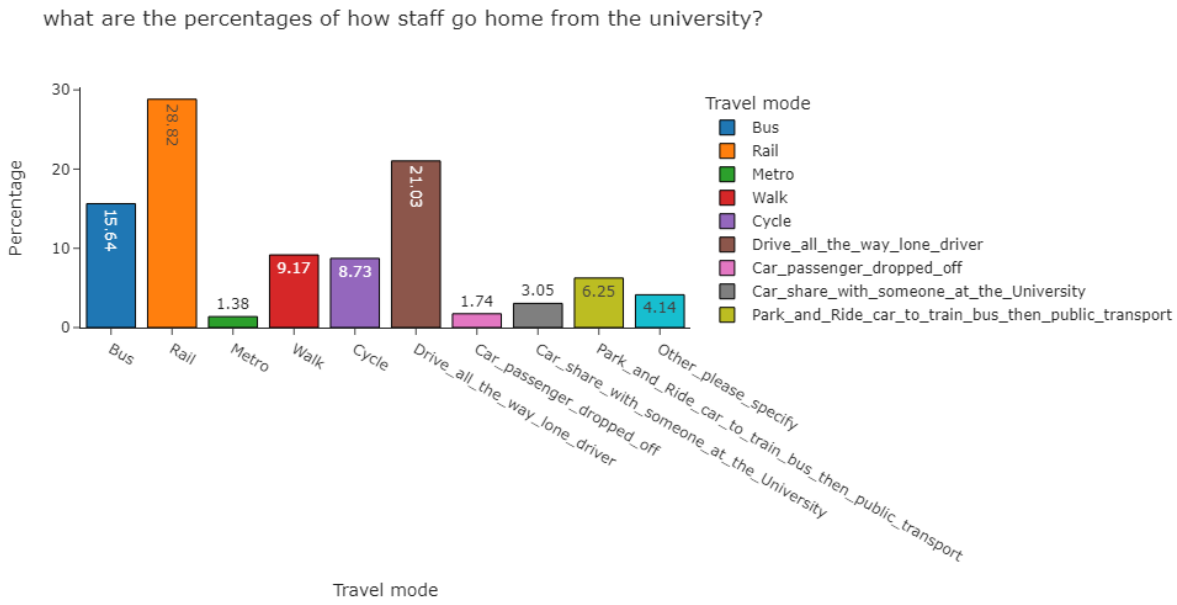

Figure 5: percentages of staff travel from the university by various modes

This plot shows the number of students and staff who gave responses to the commuters survey. 431 staff and 324 students filled out this survey. A total of 755 people responded to this survey from this graph.

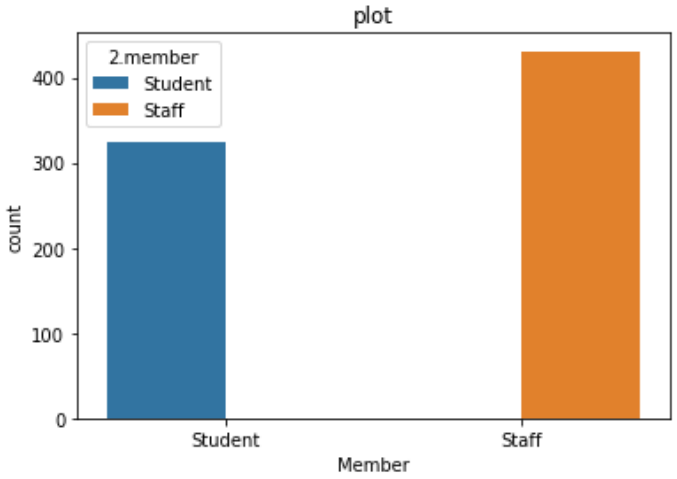

Figure 6: plot for how many students and staff are in this survey data

Travel behaviour depends on gender. In our dataset, we have 498 females, 249 males and 8 prefer not to say. This plot represents the relationship between members (students & staff) and gender features. This graph is showing how many males and female students and staff are in this survey. The below table clearly shows how many students and staff are male or female.

Gender	Student	Staff
Female	226	272
Male	96	153
Prefer not to say	2	6

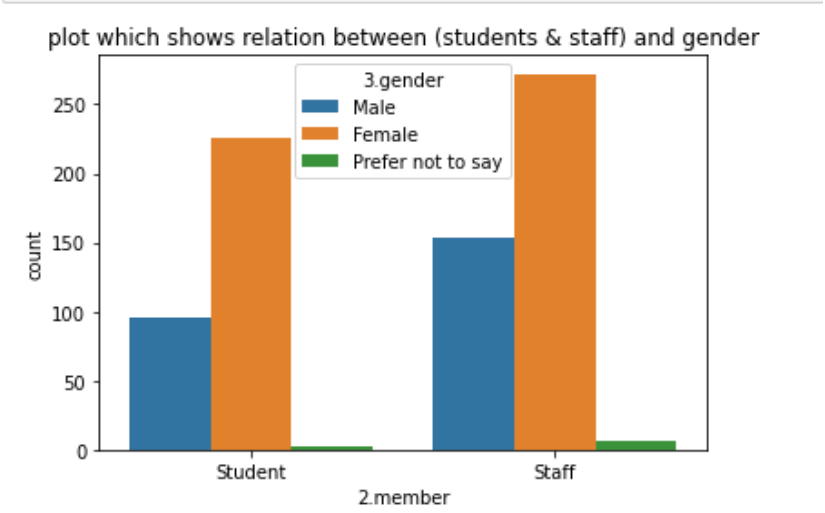

Figure 7: plot for member and gender

This plot is showing the people who responded to this survey belong to which age group people.

Age group	Student	Staff
16-24	299	29
45-59	6	149
35-44	2	119
25-34	17	103
60+	2	27
Prefer not say	0	4

This plot is showing that most of the people are students from the 16-24 age group. This table and the plot show the relationship between members and age group, which means students and staff belong to which age group.

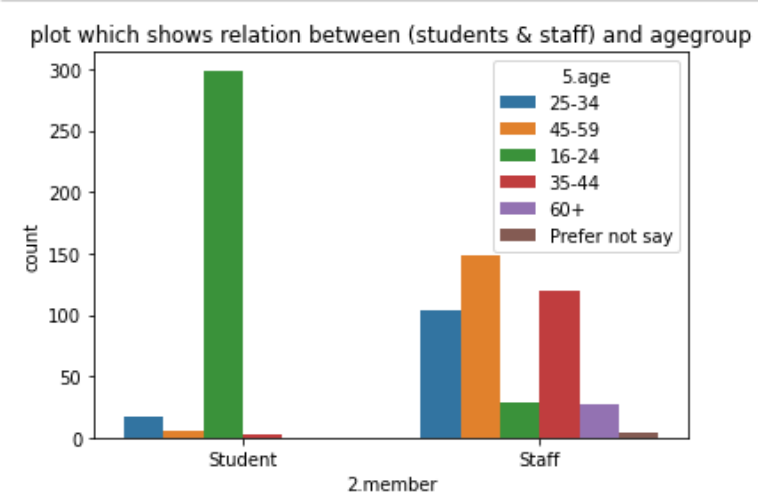

Figure 8: plot for Which age group students and staff belong to?

The travel mode behaviour sometimes depends on the distance from the house to the workplace. Because based on distance, few people choose their travel mode. Here, this plot shows how much distance it will take from the house to the university for both students and staff.

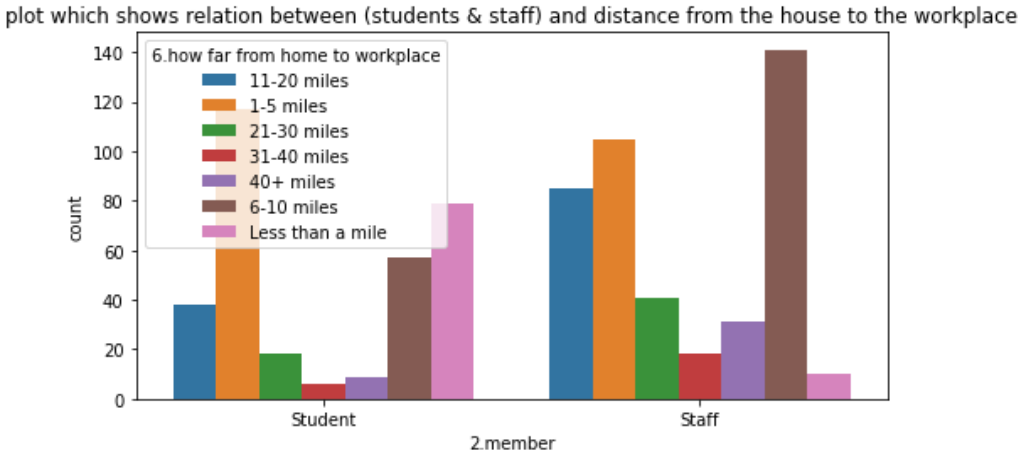

Figure 9: plot for Approximately how far is it to the University from your term time home address?

The travel mode behaviour sometimes depends on the mode of travel changes between Summer and Winter. The travel behaviour depends on the weather. Most of the students and staff responded they are not changing their travel between summer and winter.

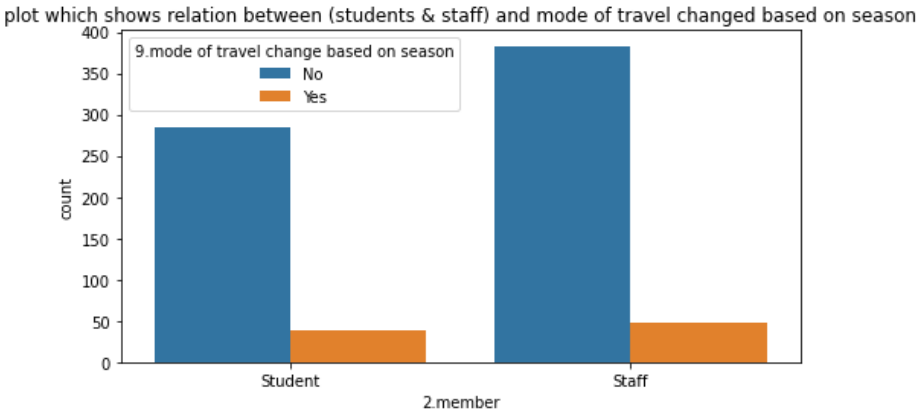

Figure 10: plot for Does the mode of travel you use change between Summer and Winter?

This plot shows, why students and staff choose the travel the way they do. Most people were chosen because those are the quickest way to reach the university.

Quickest way 262, Cheapest way 141, No alternative 88, Other (please specify) 75, Gives flexibility 63, Health reasons 31, Environmental reasons 28, Reliability 27, Safest way 23, No public transport near your home 14, Need a car/van during the day 3.

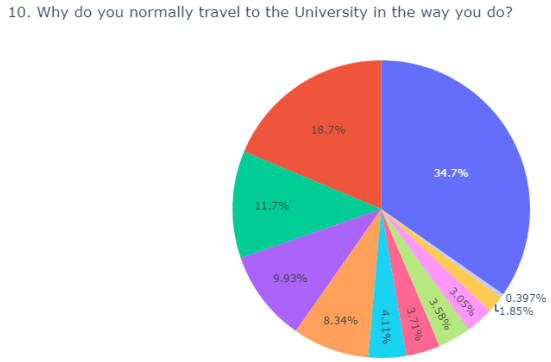

Figure 11: plot for Why do students and staff normally travel to the University in the way you do?

We have a feature called '14. How often do you travel to the University?', which represents how many times students and staff travel to the university. From this plot, most of the staff are travelling 5 times per week to the university. 174 students are travelling 5 times per week to the university.

how many times travel to the university	Student	Staff
5 times per week	174	317
3- 4 times per week	129	96
1-2 times per week	15	14
Once a month	4	0
Every couple of weeks	0	3
Less frequent than once a month	2	1

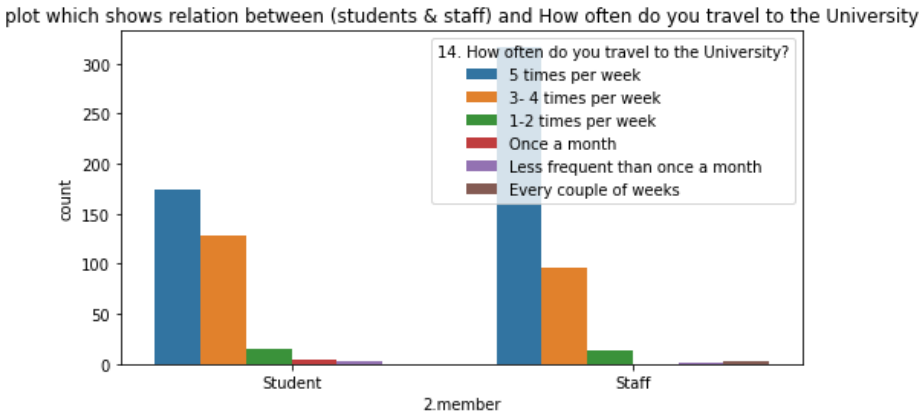

Figure 12: plot for how often students and staff travel to the university

These are the graphs which explain the demographic features of both students and staff.

4.4 Data preparation

4.4.1 Dealing with missing values

Dealing with missing data is crucial to preventing biased outcomes. Missing data in the training dataset often reduces the model's power and give results in incorrect prediction or classification. In our dataset, missing the data is not random. The survey consists of

personal opinions or decisions about questions. Generally, few people don't want to reveal about them. Like, as which age group they belong to, whether they are male or female, etc. In these cases, it is difficult to handle missing values. The handling of missing values in survey data is difficult. We don't have a particular method to handle missing data in surveys. In this study, we are handling missing values using a few methods but these are not exact values. The data analysts who handle missing values are not sure about whether those methods give particular results or not.

Here we are using a machine learning model to impute missing values in the columns. Before imputing the missing values in these columns, our data should be numerical. The machine learning model will not understand the data which have categorical variables. Our dataset consists of categorical data. So, we must convert these categorical data types of features into numerical ones. The label encoder() is used to convert these categorical features into numerical ones. This label encoder encodes the data between 0 and the number of classes in a column minus 1. If the class repeated in that column, it would assign the same number for that class. We are using this label encoder to encode each column separately.

Here, the Random Forest model is used to fill in the missing values in a few columns. They are 12. reasons for the change of mode of transport used on your daily journey to the university over the past 2 years, 18. need a car for other purposes, '19. If you travel by car to the University, where do you normally park?', '20. If you park the car within the campus, have you any difficulty finding a parking space?', 21. What would you do if car parking was unavailable at the University?', '22. Are you aware of the Aston University car share scheme?', '23. Which of the following would encourage you to car share either as a driver or passenger or improvements', '31. Which of the following would encourage you to travel here more by bicycle or improvements'.

The best imputation technique to handle missing values of categorical type is Random Forest. Because Random Forest is an efficient method for imputation and it will handle mixed types of data i.e., both numerical and categorical types. Random Forests are good at scaling to large data sets and can tolerate outliers and non-linearity in the data. They have built-in feature selection methods [9].

We are predicting missing values of 8 columns using this model. For every column, we must repeat the below procedure. Remove all the columns containing Nan values except the predicted missing ones in the column. Then separate the null and non-null values from the dataset. Consider null values as test data and nonnull values as the training dataset. Select all the features except predicting column from the training dataset as the x_train_feature number and drop all features except predicting column from the training dataset as the y_train_feature number. Based on training data, fit (x_train_feature number, y_train_feature number) and build the Random Forest model. Create an x_test_feature number from the test dataset except predicting column. Apply the model on the x_train_feature number of test data and predict the y_train_feature number. Replace these missing values with predicted values in that column of the original dataset. This is the procedure using a machine learning algorithm to handle missing values in the data. Repeat this procedure for those columns are having missing values.

4.4.2 Feature selection

Now, it's time to deal with feature selection. We have irrelevant features in our dataset, which are not useful for our prediction. So, remove those columns or select the required features for prediction. Here, we consider demographic data as independent variables and travel modes as dependent variables. The features which are '2.member', '3.gender', '5.age', '6.how far from home to workplace', '9.mode of travel change based on season', '10. Why do you normally travel to the University in the way you do?', '14. How often do

you travel to the University?' and '17.journey time to the university' etc. are independent variables. And '7.5.f. Bus - Everyday', '7.2.f. Rail - Everyday', '7.3.f. Metro - every day', '7.4.f. Walk - Everyday', '7.5.f. Cycle - Everyday', '7.6.f. Drive all the way (lone driver) - every day', '7.7.f. Car passenger (dropped off) - every day, '7.8.f. Car share (with someone at the University) - every day', '7.9.f. Park and Ride (E.g. car to train/bus then public transport) - every day', '7.50.f. Other (please specify) - every day', '8.1.f. Bus - Everyday', '8.2.f. Rail - Everyday', '8.3.f. Metro - every day', '8.4.f. Walk - Everyday', '8.5.f. Cycle - Everyday', '8.6.f. Drive all the way (lone driver) - every day', '8.7.f. Car passenger (dropped off) - every day, '8.8.f. Car share (with someone at the University) - every day', '8.9.f. Park and Ride (E.g. car to train/bus then public transport) - everyday' and '8.10.f. Other (please specify) - every day' etc. are target variables or dependent variables.

Here, we predict the travel mode behaviour to the university by students and staff and the travel mode behaviour from the university by students and staff separately. To predict the travel mode behaviour to the university by students and staff consider these features as dependent variables such as '7.5.f. Bus - Every day', '7.2.f. Rail - Every day', '7.3.f. Metro - Every day', '7.4.f. Walk - Every day', '7.5.f. Cycle - Every day', '7.6.f. Drive all the way (lone driver) - Every day, '7.7.f. Car passenger (dropped off) - Every day, '7.8.f. Car share (with someone at the University) - Every day, '7.9.f. Park and Ride (E.g. car to train/bus then public transport) - Every day, '7.50.f. Other (please specify) - Every day. Generally, the models have only one output. Whereas we have multiple outputs. So, we are using a Multioutput classifier() ensemble with a Random forest model.

To predict the travel mode behaviour from the university by students and staff consider these features as dependent variables such as '8.1.f. Bus - Everyday', '8.2.f. Rail - Everyday', '8.3.f. Metro - Every day', '8.4.f. Walk - Everyday', '8.5.f. Cycle - Everyday', '8.6.f. Drive all the way (lone driver) - Every day, '8.7.f. Car passenger (dropped off) - Every day, '8.8.f. Car share (with someone at the University) - Every day, '8.9.f. Park and Ride (E.g. car to train/bus then public transport) - Every day and 8.10.f. Other (please specify) - Every day etc. are target variables or dependent variables. So, we are using a Multioutput classifier() ensemble with a Gradient boosting classifier().

4.5 Machine learning models

After feature selection, we build the machine learning models. These models are used to train the model and predict the target features. These models can help us to take better decisions and predictions of business problems. We can predict future trends using previous data or historical data regarding business problems.

4.5.1 Multioutput classifier

Generally, in machine learning, every model predicts single output after prediction. In our case, we have to predict multiple outputs. To predict multiple outputs we use a Multioutput classifier. This multioutput classifier gives more than one output after prediction simultaneously. In this study, we are predicting 10 features which represent the travel modes to the university and 10 features which represent the travel mode from the university by students and staff.

The key advantage of adopting Random Forest and Gradient boosting over traditional statistical methods is that they deal with irrelevant factors considerably better. Still, feature engineering is a component of feature selection, which is still valuable and important. If we wish to utilize our model again, we should still strive to limit the number of predictors for practical reasons. Because every predictor employed in the training model must also be present in the unseen data, regardless of how irrelevant they may be. Tree-based methods (decision trees, Random Forest) can work with categorical data and label encoding.

4.5.2 Decision Tree[18]

A decision tree is a supervised machine learning, which is represented as a tree structure and used to mention the course of actions. This algorithm is used for both classification and regression. This model consists of branches that represent possible occurrences or decisions and nodes. These nodes can be split into 3 nodes. They are the parent node(further divided into two or more homogeneous sets), the decision node(split into sub-nodes) and the leaf node(not split further). The decision tree follows an if-else structure, every node uses one independent feature to split into two or more branches.

To decide whether to divide a node into two or more sub-nodes, decision trees employ a variety of techniques. The homogeneity of newly formed sub-nodes is increased by sub-node formation. In other words, we may say that the node's purity about the target variable grows. The decision tree divides the nodes based on all of the factors that seem to be accessible, and then chooses the split that produces the most homogeneous sub-nodes. If our independent features are categorical, these categories are used to decide the split of the nodes. Among the most widely used algorithms in the decision tree, is the entropy algorithm, which measures the impurity of a dataset. Several forms of measurement may be used to assess a dataset's impurity. The Gini Index, Chi-Square, Information Gain Ratio, and Variance are the most notable ones.

Entropy:

The entropy of a dataset is used to measure the impurity of a dataset, therefore it is one of the most commonly used algorithms in the decision tree. The definition of information entropy is general and expressed in terms of a discrete set of probabilities P so that

$$E(x) = -\sum_{k\in \text{ target}} P(x = n)\log_2(P(x = n)).$$

where $P(x = n)$ is the probability that the target feature takes a specific value n.

After calculating entropy, we need to find which feature splits the data along the target variable values accurately in terms of information gain which should be used to split the data on the parent node.

This approach, known as the Information Gain algorithm, uses entropy as its primary principle to help identify a feature or attribute that provides the most information about a class. This approach allows us to lower the entropy from the root node to the leaf node. Based on the attribute values, information gain calculates the difference between average entropy before the split and before the division of the dataset. Entropy is only a measurement of disorder.[18] The information gained is defined as

Information Gain (feature_d) = Entropy (Data) - Entropy (feature_d)

Information Gain or a minimal reduction in impurities from the root/parent node to the child nodes. Calculate Information Gain for every independent feature and check which feature has more Information Gain. That feature is used to split the data.

Whereas in python, by default it will use Gini impurity. In our project, we used the Random forest to deal with missing values. Here we are taking a subset of our dataset. This is only for understanding purposes. Because we applied these models after label encoding on our data. After encoding, our data is numerical.

10. Why do you normally travel to the University in the way you do?	11. How would you prefer to travel to work?	18. need a car for other purposes
Reliability	Drive all the way-lone	No
Quickest way	Drive all the way-lone	No
Gives flexibility	Drive all the way-lone	Yes
Quickest way	Drive all the way-lone	
Gives flexibility	Drive all the way-lone	

Figure 13: Example data

Separate nonnull data as training and null as test data. Predict missing values using training data. Here, we are predicting missing values of '18. need a car for other purposes feature' using '10. Why do you normally travel to the University in the way you do?' and '11. How would you prefer to travel to work?' as independent variables. Let's see how it works

Gini impurity:

Gini impurity is a measurement of the likelihood that a specific variable would be incorrectly categorised when it is randomly selected. It is a statistic to measure how frequently a randomly selected feature might be mistaken for another. The Gini index indicates that the data are evenly distributed.[18]
The Gini index is measured as the sum of the squared probabilities of each class in an independent feature.

$$\text{Gini impurity} = 1 - \text{ Gini index}$$

$$\text{Gini impurity} = 1 - \sum_{i=1}^{n} (P_i)^2$$

P_i = probability of an item being classified into a particular class
calculating the Gini index for '10. Why do you normally travel to the University in the way you do?',
P(10. Why do you normally travel to the University in the way you do? = Reliability) = 1/5
P(10. Why do you normally travel to the University in the way you do? = Quickest way) = 2/5
P(10. Why do you normally travel to the University in the way you do? = Gives flexibility) = 2/5

If (10. Why do you normally travel to the University in the way you do? = Reliability & 18. need a car for other purposes = No), probability = $\frac{1}{5}$
If (10. Why do you normally travel to the University in the way you do? = Reliability & 18. need a car for other purposes = Yes), probability = 0
If (10. Why do you normally travel to the University in the way you do? = Quickest way & 18. need a car for other purposes = No), probability = $\frac{1}{5}$
If (10. Why do you normally travel to the University in the way you do? = Quickest way & 18. need a car for other purposes = Yes), probability = 0
If (10. Why do you normally travel to the University in the way you do? = Gives flexibility& 18. need a car for other purposes = Yes), probability = $\frac{1}{5}$

If (10. Why do you normally travel to the University in the way you do? = Gives flexibility& 18. need a car for other purposes = No), probability = 0

$$\text{Gin impurity } = 1 - \left(\left(\frac{1}{5}\right)^2 + 0 + \left(\frac{1}{5}\right)^2 + 0 + \left(\frac{1}{5}\right)^2 + 0\right) = 1 - 0.12 = 0.88$$

Calculate the Gini impurity for every individual feature.

The entropy is in the range between 0 to 1 and Gini Impurity lies in the range between 0 to 0.5. But the Gini Impurity is better than entropy for selecting the best features.[18] we used Gini Impurity to select the best features or to spit the decision tree. And also if you use Entropy, it will take more time.

4.5.3 Ensemble methods[6]

The ensemble methods are used to improve the accuracy of results in models by combining various models instead of using one model. The combined various models increase the accuracy of the results. Ensemble methods have two categories. They are sequential ensemble techniques, which generate base learners in sequence and parallel ensemble techniques, which generate base generators in parallel[6]. The Ensemble methods are used mostly as Bagging, Boosting and stacking.

Bagging:

The bagging method is mostly used in regression and classification. These ensemble methods are used to improve the model's accuracy and reduce variation. A few machine-learning models struggle with overfitting. This problem can be eliminated by increasing accuracy and minimising variance[6].

The sampling with replacement approach aids in the randomization of the selection process. The process is finished by applying the basic learning algorithm to the samples.

The bagging method consists of two types. Bootstrapping, which is a technique used to derive samples from a set using replacement and aggregation is used to include all potential outcomes of the prediction and randomize the result in bagging. The sampling with replacement approach results in the randomization of the selection process. Without aggregation, predictions are not accurate. The aggregation is based on the probability of bootstrapping techniques[6].

Bagging methods are used to create a strong learner which is stronger than a single weak base learner. The drawback of this bagging method is computationally expensive. If we ignore the correct process of the bagging technique it can lead to more bias in models. Random Forest is a powerful model and it is a type of ensemble method called bagging[6].

Boosting:

Boosting is an ensemble method which learns from previous predictor errors and improves future predictions. This method combines weak-base learners into a single strong learner to increase model predictability. Boosting method placing weak learners in sequential so that they can learn from the subsequent learner to improve their prediction models[6].

Gradient boosting adds predictors to the ensemble sequentially and allows forecasters to correct previous ones. To handle incorrect accuracies in previous predictors, new predictors are fitted. This method identifies the problems and corrects the issues of learners according to the gradient of descent[6].

4.5.4 Random Forest classifier[16]

A machine learning technique called random forest or random decision forest creates several decision trees during the training stage. The random forest makes the ultimate

choice using the majority decision of the trees. It is a type of ensemble learning in which several weak models are combined to create a strong model. (Randa class)

A machine learning technique which is used mostly to solve both classification and regression problems is known as Random Forest Algorithm. This model is built with various kinds of trees. If the model uses more number decision trees and the model will be more robust. The accuracy and problem-solving capacity of this algorithm increase with the number of decision trees in the algorithm. So, the number of trees and the accuracy of the model are directly proportional to each other. If we want to increase the prediction accuracy of the data this classifier is used and it uses several decision trees on different subsets of training data or input information. It is based on the concept of ensemble learning, which is the process of combining several classifiers to address a complex problem and improve the performance of the model[19].

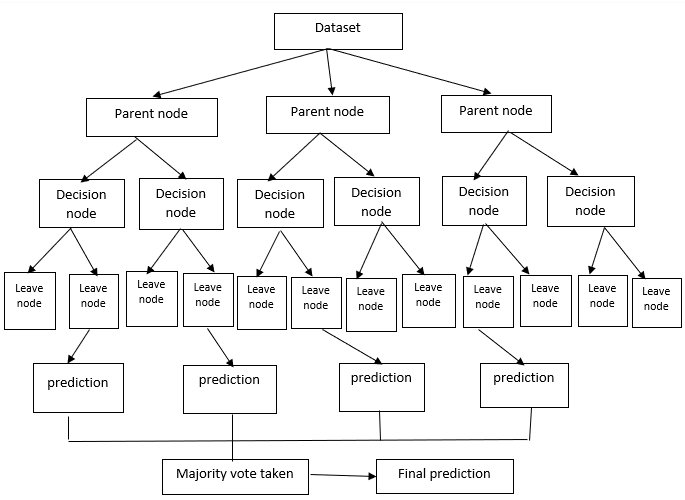

Figure 14: Random forest structure

Each tree is grown based on the below steps. The steps that take place in this algorithm are:

- First, this algorithm selects random samples from the given input data or training dataset.
- It will build a decision tree for every training dataset.
- Average the decision trees then voting will take place based on average.
- predicts the new dataset based on the majority votes for the classification of the prediction of the decision trees.

The Random forest model makes weak base learners combined to form a single strong learner and give good accuracies. In this model, we created base learners in a parallel format. To promote independence among the base learners, parallel techniques make use of parallel generations of base learners.

In the example mentioned above in decision trees, this model also builds a various number of trees. Each tree predicts outputs and voting will take place after the prediction.

4.5.5 Gradient Boosting classifier[13][15]

One of the variations of ensemble techniques is gradient boosting, which involves building numerous weak models (typically decision trees) and combining them to improve performance overall. If the target feature is not continuous we use a Gradient Boosting classifier.[13]

There are four steps involved in the Gradient boosting algorithm. The first step is to create a model with a constant value.

$$F_0(x) = \arg_\gamma \min \sum_{i=1}^{n} L\left(y_i, \gamma\right)$$

Here, our problem is classification. So, log-likelihood is the loss function for classification. The loss function for the classification problems is given by

$$L = -\left[\sum_{i=1}^{n} y_i \log(p) + (1 - y_i) \log(1 - p)\right]$$

where ' y ' is the target, it is either 0 or 1. ' P ' is the predicted probability of class 1.

$$L = \begin{cases} -\log(p), & \text{if } y_i = 1 \\ -\log(1-p), & \text{if } y_i = 0 \end{cases}$$

according to above function, if increase p for $y_i = 1$, then the smaller loss.
we have to minimize the loss using gamma or predicted value. Find the γ' value.
Assume γ' as log-odds. we are converting ' P ' into log-adds and adding γ'.

$$\log\left(\, odds\right) = \log\left(\frac{p}{1-p}\right)$$

To solve 'argmin' in terms of log(odds):

$$\begin{aligned} L &= -\left[\sum_{i=1}^{n} y_i \log(p) + (1 - y_i) \log(1 - p)\right] \\ &= -\left[\sum_{i=1}^{n} y_i \log(p) + \log(1 - p) - y_i \log(1 - p)\right] \\ L &= -\left[\sum_{i=1}^{n} y_i(\log(p) - \log(1 - p)) + \log(1 - p)\right] \end{aligned}$$

we know that $\log a - \log b \log \frac{a}{b}$

$$\begin{aligned} L &= -\left[\sum_{i=1}^{n} y_i \log\left(\frac{p}{1-p}\right) + \log(1 - p)\right] \\ L &= -\left[\sum_{i=1}^{n} y_i \log(\text{odds}\, s) + \log(1 - p)\right] \rightarrow (1 \end{aligned}$$

we know that, log (odds)

$$\begin{aligned} &\log(\text{ odds }) = \log\left(\frac{p}{1-p}\right) \\ &e^{\log(0dds)} = \frac{p}{1-p} \\ &(1-p)e^{\log(0ddS)} = p \\ &e^{\log(\text{ odds })} - pe^{\log(\text{ odd })} = p \\ &e^{\log(0dds)} = p + pe^{\log(0dds)} \\ &e^{\log(0dds)} = P\left(1 + e^{\log(0dds)}\right) \\ &\frac{e^{\log(0dds)}}{1 + e^{\log(0dds)}} = P \end{aligned}$$

substitute the above ' p ' value in Eqn (1)

$$L = -\left[\sum_{i=1}^{n} y_i \log(0ddS) + \log\left(1 - \frac{e^{\log(odds)}}{1 + e^{\log(0dds)}}\right]\right.$$

$$L = -\left[\sum_{i=1}^{n} y_i \log((\ odds) + \log\left(\frac{1 + e^{\log(\text{odds}}) - e^{\log(\text{odds})}}{1 + e^{\log \text{odds})}}\right]\right.$$

$$= -\left[\sum_{i=1}^{n} y_i \log(0dds) + \log(1) - \log\left(1 + e^{\log(\text{lodd } s)}\right]\right.$$

we know that $\log \frac{a}{b} = \log a - \log b$.

$$L = -\left[\sum_{i=1}^{n} y_i \log(\text{lod } ds) - \log\left(1 + e^{\log \text{odds}}\right)\right)\right]$$

Derivative of ' L ' with respect to log odds.

$$\frac{\partial(L)}{\partial \log(\ \text{odds}\ )} = \frac{\partial}{\partial \log(0dds)}\left(-\left[\sum_{i=1}^{n} y_i \log(0\text{dds}) - \log\left(1 + e^{\log(\text{odds})}\right)\right]\right.$$

$$= \frac{\partial}{\partial \log(0dds)}\left(-\sum_{i=1}^{n} y_i \log(0dds) + \log\left(1 + e^{\log(\text{odd } s)}\right)\right)\right)$$

$$= -\sum_{i=1}^{n} y_i + n\frac{1}{1 + e^{\log(0ddS)}} e^{\log(0ddS)}$$

$$= -\sum_{i=1}^{n} y_i + n\frac{e^{\log(0dds)}}{1 + e^{\log(0dds)}}$$

$$= -\sum_{i=1}^{n} y_i + np.$$

$$\frac{\partial k}{\partial \log(0ddS)} = 0$$

$$-\sum_{i=1}^{n} y_i + np = 0$$

We are solving this one for 'p'.

$$np = \sum_{i=1}^{n} y_i$$

$$p = \sum_{i=1}^{n} \frac{y_i}{n}$$

$$P = \frac{1}{n}\sum_{i=1}^{n} y_i$$

$$P = \bar{y}$$

here $\bar{y}$ is the mean of y is the proportion of class 1 . Then,

$F_0(x) = \log(0dds) = \gamma = \log\left(\frac{\bar{y}}{1-\bar{y}}\right)$

$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for $i = 1, \ldots n$

where $F\left(x_i\right)$ is the previous model and m is the number of Decision Tree.

The second step in the Gradient boosting classifier is to compute residuals.

$$\begin{aligned}\gamma_{im} &= -\frac{\partial(L)}{\partial \log(\text{ odds })} \\ &= -\left[-y_i + p\right] \\ \gamma_{im} &= y_i - p\end{aligned}$$

Here we used a negative gradient which gives the directions and magnitude in the loss function that can be minimized.
The third step is to find the output values for each leaf of the decision tree.
Train the regression tree. In this step find the output values for each leaf in the decision tree. To find output, take an average of all numbers in a leaf.

$$\begin{aligned}r_m &= \arg_\gamma \min \sum_{i=1}^{n} L\left(y_i, F_{m_1}\left(x_i\right) + \gamma\right) \\ &= \text{argmin}_\gamma \sum_{i=1}^{n} - \left[y\left(F_{m_{-1}}\left(x_i\right) + \gamma\right) - \log\left(1 + e^{F_{n-1}}\left(x_i\right) - \gamma\right)\right] \text{ as}\end{aligned}$$

we know that second-order Taylor polynomial is

$$f(x) = f(a) + f'(a)(x-a) + \frac{1}{2}f''(a)(x-a)^2$$

$$L\left(y, F_{m-1}\left(x_i\right) + \gamma\right) \approx L\left(y_1 F_{m-1}\left(x_i\right)\right) + \frac{\partial}{\partial F} L\left(y_i, F_{m-1}\left(x_i\right)\right)\gamma + \frac{1}{2}\frac{\partial^2 L\left(y_i, F_{m-1}\left(x_i\right)\right)}{\partial F^2}\gamma^2$$

derivate the above equation with respect to γ and equal to zero

$\frac{\partial}{\partial\gamma} x_i \in R_{jm} \cdot \left(L\left(y_i, F_{m-1}\left(x_i\right)\right) + \frac{\partial}{\partial F} L\left(y_i, F_{m-1}\left(x_i\right)\right)\gamma + \frac{1}{2}\frac{\partial^2 L(y_i, F_{m-1}(x_i))}{\partial F^2}\gamma^2\right) = 0$

$\sum_i \in R_{jm}\left(\frac{\partial}{\partial F} L\left(y_i, F_{m-1}\left(x_i\right) + \frac{\partial^2 L(y_i, F_{m-1}(x_i))}{2\partial F^2}(2\gamma)\right)\right) = 0$

$\sum_{x_i \in R_{jm}} \frac{\partial^2 L(y_i, F_{m-1}(x_i))}{\partial F^2}\gamma = -\sum_{x_i \in R_{jm}} \frac{\partial}{\partial F} L\left(y_i, F_{m-1}(x_i)\right)$

$$\gamma^2 = \frac{-\sum_{x_i \in R_{jm}} \frac{\partial}{\partial F} L\left(y_i, f_{m-1}\left(x_i\right)\right)}{\sum_{x_i \in R_{jm}} \frac{\partial^2}{\partial F^2} L\left(y_i, F_{m-1}\left(x_i\right)\right)}$$

as we already know,

$$\begin{aligned}&\frac{\partial L\left(y_i, f\left(x_i\right)\right)}{\partial f\left(x_i\right)} = -\left(y_i - p\right) \\ &\text{then } \gamma = -\frac{\sum_{x_i \in R_{jm}} -\left(y_i - p\right)}{\sum_{x_i \in R_{jm}} \frac{\partial}{\partial F}\left(-\left(y_i - p\right)\right)} \\ &\gamma = \frac{\sum_{x_i \in R_{jm}} y_i - p}{\sum_{x_i \in R_{jm}} \frac{\partial}{\partial f}\left(-y_i + \frac{e^{\log(0dds)}}{1 + e^{\log(0dds)}}\right)} \\ &\frac{\partial}{\partial F}\left(-y_i + \frac{e^{\log(0dds)}}{1 + e^{\log(0dds)}}\right) = 0 + \frac{\partial}{\partial f}\left(e^{\log(0dds)}\right)^{-9} \\ &= e^{\log(\text{ odd })}\left(1 + e^{\log(odds))}\right)^{-1} - e^{2\log(\text{ odd })}\left(1 + e^{\log(odds))}\right)^{-2} \\ &= \frac{e^{\log(0dds)}}{1 + e^{\log(0dds)}} - \left(\frac{e^{\log(0dds)}}{1 + e^{\log(0dds}}\right)^2\end{aligned}$$

substitute this value in the above ' γ ' equation.

$$\gamma = \frac{\sum_{i \in R_{jm}} \dot{y}_i - p}{\sum_{x_i \in R_{jm}} \cdot p - p^2}$$

$$\gamma = \frac{\sum_{x_i \in R_{jm}} (y_i - p)}{\sum_{x_i \in R_{jm}} P(1-P)}$$

The final step is to update the predictions of the previous model as

$$F_m(x) = F_{m-1}(x) + V_{j=1}^{\sum_j^{Im} \gamma_{jm}1} (x \in R_{jm})$$

These are the steps for the Gradient boosting algorithm, to get the model performance we have to iterate the second step M times. This means this one adds M trees to the combined model[13]. If you want to get the best model then you have to add nearly 100 trees.

5 Evaluation

The below table represents the percentages of students and staff that comes by bus, rail, metro, walk, cycle, drive all the way, car share, park and ride (E.g., car to train/bus then public transport), other etc from house to the university and university to the house is listed.

Travel Modes	Travel to the university		Travel from the university	
	Students	Staff	Students	Staff
Bus percentage	30.61%	15.81%	25.56%	15.64%
Rail percentage	17.55%	26.60%	19.65%	28.82%
Metro percentage	2.67%	2.18%	2.34%	1.38%
Walk percentage	33.62%	10.22%	36.08	9.17%
Cycle percentage	2.89%	8.08%	2.43%	8.73%
Drive all the way (lone driver) percentage	4.92%	21.71%	5.39%	21.03%
Car passenger (dropped off) percentage	2.84%	2.57%	4.69%	1.74%
Car share (with someone at the University) percentage	0.54%	3.53%	0.60%	3.05%
Park and Ride (E.g. car to train/bus then public transport) percentage	3.28%	7.03%	2.34%	6.25%
Other (please specify) percentage	1.03%	2.22%	0.86%	4.14%

From the above table, we can say that students are mostly preferred to walk whereas staff are preferred to come by car. Observe these percentages. From this, we can understand staff need more improvements regarding travel.

This study attempted to understand how students and staff currently travel to/from the university. An online questionnaire survey is conducted at Aston university, but 755 responses from the survey. Using this survey data with statistical methods to understand the behaviour of students and staff, their attitudes and their mode of transportation. In methodology, the analysis of data which is both graphical and statistical analysis is done. From that analysis, we can clearly understand the relation between dependent features.

The staff travel to/from the university using cars. Then they used rail and bus. We have to be concerned about the staff. Because day by day our earth losing a healthy environment. So, try to reduce cars, and increase eco-friendly vehicles. Encourage youth to use cycles or which reduces pollution.

From this project, we are predicting which factors are affecting travel mode. Why people are choosing the way they travel to the university and what encourages the students and staff to choose those travel modes? To do this analysis, we selected a few features which are affecting the travel modes in feature selection. In our methodology, we selected these features affecting travel mode which are '2.member', '3.gender', '5.age', '6.how far from home to workplace', '9.mode of travel change based on season', '10. Why do you normally travel to the University in the way you do?', '14. How often do you travel to the University?' '17.journey time to the university' etc. To build a strong model, we used a Random Forest classifier() and Gradient Boosting classifier(). The Random Forest classifier is used to predict which features are affecting the student's and staff's travel to

the university. This model reduces variance and overfitting in the model. By using this model, we got good accuracies.
7.5.f. Bus - Everyday 75.77
7.2.f. Rail - Everyday 75.33
7.3.f. Metro - Everyday 95.59
7.4.f. Walk - Everyday 85.02
7.5.f. Cycle - Everyday 89.86
7.6.f. Drive all the way (lone driver) - Everyday 80.61
7.7.f. Car passenger (dropped off) - Everyday 95.15
7.8.f. Car share (with someone at the University) - Everyday 96.91
7.9.f. Park and Ride (E.g. car to train/bus then public transport) - Everyday 92.51
7.50.f. Other (please specify) - Everyday 96.47

We used the Random model to predict missing values and target features. The Random forest classifier combines weak learners into one single strong learner here and these learners are generated in parallel format[6]. To evaluate whether our model is working well or not, we calculate accuracy, precision and recall. If accuracy is good, we can understand our model is working well and predicting results correctly. From these accuracies, we can say that travel mode behaviour is affected by these demographic features of people who travel to the university.

The Gradient Boosting classifier is used to predict which features are affecting the students and staff's travel from the university.
Here this model makes weak base learners into single strong learner and add predictors to the ensemble sequentially. So, this model gives good accuracy.
8.1.f. Bus - Everyday 75.33
8.2.f. Rail - Everyday 72.24
8.3.f. Metro - Everyday 95.59
8.4.f. Walk - Everyday 75.77
8.5.f. Cycle - Everyday 90.74
8.6.f. Drive all the way (lone driver) - Everyday 75.33
8.7.f. Car passenger (dropped off) - Everyday 94.71
8.8.f. Car share (with someone at the University) - Everyday 97.35
8.9.f. Park and Ride (E.g. car to train/bus then public transport) - Everyday 93.39
8.10.f. Other (please specify) - Everyday 91.62

From these accuracies, we can understand this Gradient boosting model is also working well. We have target features that are not continuous so we used Gradient boosting classifier. Our problem is a classification problem. These two models, the Random forest classifier and Gradient boosting classifier models are used to get the best model performance. We have multiple outputs in our case so we used a Multioutput classifier ensemble with these two models. Generally, this multioutput classifier will not support any metrics. So we calculated accuracy without using metrics. We can see our model's accuracy is good.

There are a few questions for staff and students which are used to know about their travel. Like why do they travel how the way they travel to/from the university, most of the staff and students gave those are the quickest way. If they had changed their travel mode then how did they travel in the last two years and why? to this most of the people responded that they moved house, Financial reasons including changes to the cost of travel or parking and Other (change in circumstances, the unreliability of public transport). And what would encourage them to use public transport are discounts on public transport tickets/passes. What would they do if car parking was unavailable at the university, for this most of the students and staff choose still travel by car but park

elsewhere. From these responses, we can understand still staff and students are interested in coming by car if parking is not available at the university.

From these responses, the government has to understand the problems behind travel behaviour and take certain measures to reduce pollution. To guarantee that parking levels are compatible with the continued transition toward active, sustainable, shared, affordable, and efficient means of transportation, this strand also covers the suggested method of providing automobile parking for visitors, the general public, employees, and students.

6 Conclusion

From the methodology and evaluation, the demographic features of people such as member(student or staff), age group, gender(male or female), distance from the home to the workplace of them, and mode of travel changed between summer and winter, why do they travel to the university in the way they do, how often they travel to the university, and how much time it will take from the university. These are the factors that affect travel mode behaviour. The distribution of how much percentages of people get into the campus, if you observe this distribution we will know percentages of travel modes. Based on these percentages, try to improve transport facilities and encourage which mode of travel is eco-friendly with nature.

The surveys also make it clear that the expense of public transport and the network of cycle paths that provide access to the site are major obstacles to raising the modal shares of both modes. According to the study, there is still work to be done at the university to improve facilities, such as bike parking and showers, as well as other initiatives like flexible work schedules for employees and financial assistance for bike purchases through a cycle-to-work programme.

Aston university has welcomed the Clean Air Zone(CAZ) which tackles air pollution in Birmingham city. This program leads to improve air quality and a healthy environment for walking and cycling in the city centre. Most of the staff are coming to/ from the university by car, which increases carbon emissions on the campus. So try to improve transport facilities and provide discounts on public transportation. Why do we need to do these means, Aston University actively promotes sustainable travel and was recently recognised for its sustainable travel strategy by Mode shift stars to help Birmingham achieve its 2030 net-zero goal and lower the number of automobiles entering the city centre. There are benefits of sustainable travel like using public transportation reduces pollution and congestion, and travel modes such as cycling and walking are good for health. These sustainable travel options save money on the rising cost of fuel.

Our vision states the goal of creating a network that is intelligent, inventive, carbon-neutral, and low-emission, which will enable inclusive, sustainable economic growth, tackle climate change and advance the health and wellness of its citizens. To reduce travel by car, try to decrease parking space at the university and in the city centre. If we reduce these carbon emissions, this leads to a healthy environment and those people, who live near the city centre live happily and healthily. We can reduce health diseases like lung cancer, obesity etc.

References

[1] Khondhaker Al Momin, Saurav Barua, Omar Faruqe Hamim, and Subrata Roy. Modeling the Behavior in Choosing the Travel Mode for Long-Distance Travel Using Supervised Machine Learning Algorithms. *Communications - Scientific letters of the University of Zilina*, 24(4):A187–A197, oct 26 2022.

[2] Essam Almasri and Sadi Alraee. Factors Affecting Mode Choice of Work Trips in Developing Cities—Gaza as a Case Study. *Journal of Transportation Technologies*, 03(04):247–259, 2013.

[3] Khaled J. Assi, Kh Md Nahiduzzaman, Nedal T. Ratrout, and Adel S. Aldosary. Mode choice behavior of high school goers: Evaluating logistic regression and MLP neural networks. *Case Studies on Transport Policy*, 6(2):225–230, 6 2018.

[4] Muhammed Emin Cihangir Bagdatli and Fatima Ipek. Transport mode preferences of university students in post-COVID-19 pandemic. *Transport Policy*, 118:20–32, 3 2022.

[5] Catarina Cadima, Cecília Silva, and Paulo Pinho. Changing student mobility behaviour under financial crisis: Lessons from a case study in the Oporto University. *Journal of Transport Geography*, 87:102800, 7 2020.

[6] CFI CFI team. Ensemble Methods. https://corporatefinanceinstitute.com/resources/data-science/ensemble-methods/. [Online; accessed 2023-01-03].

[7] Sandip Chakrabarti. How can public transit get people out of their cars? An analysis of transit mode choice for commute trips in Los Angeles. *Transport Policy*, 54:80–89, 2 2017.

[8] Sanhita Das, Alice Boruah, Arunabha Banerjee, Rahul Raoniar, Suresh Nama, and Akhilesh Kumar Maurya. Impact of COVID-19: A radical modal shift from public to private transport mode. *Transport Policy*, 109:1–11, 8 2021.

[9] Sunil Kumar Dash. Handling Missing Values with Random Forest - Analytics Vidhya. https://www.analyticsvidhya.com/blog/2022/05/handling-missing-values-with-random-forest/, may 4 2022. [Online; accessed 2022-12-27].

[10] James DeWeese, Léa Ravensbergen, and Ahmed El-Geneidy. Travel behaviour and greenhouse gas emissions during the COVID-19 pandemic: A case study in a university setting. *Transportation Research Interdisciplinary Perspectives*, 13:100531, 3 2022.

[11] Juan de Dios Ortúzar and Luis G. Willumsen. *Modelling Transport*. Wiley, jan 3 2002.

[12] Chengxi Liu, Yusak O. Susilo, and Anders Karlström. The influence of weather characteristics variability on individual's travel mode choice in different seasons and regions in Sweden. *Transport Policy*, 41:147–158, 7 2015.

[13] Tomonori Masui. All You Need to Know about Gradient Boosting Algorithm — Part 2. Classification. https://towardsdatascience.com/all-you-need-to-know-about-gradient-boosting-algorithm-part-2-classification-d3ed8f56541e, feb 7 2022. [Online; accessed 2023-01-03].

[14] O. Petrik, J. de Abreu e Silva, and F. Moura. Stated preference surveys in transport demand modeling: disengagement of respondents. *Transportation Letters*, 8(1):13–25, 1 2016.

[15] Anshul Saini. Gradient Boosting Algorithm: A Complete Guide for Beginners. https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners/, sep 20 2021. [Online; accessed 2023-01-03].

[16] Anshul Saini. Random Forest Algorithm for Absolute Beginners in Data Science. https://www.analyticsvidhya.com/blog/2021/10/an-introduction-to-random-forest-algorithm-for-beginners/, oct 19 2021. [Online; accessed 2023-01-03].

[17] Kate Shapiro. Our Green Campus: A Car-free Campus? https://www.unlv.edu/news/article/our-green-campus-car-free-campus, may 27 2021.

[18] Shubhtripathi Shubhtripathi. Ensembling Decision Trees. https://shubhtripathi.medium.com/ensembling-decision-trees-59356e0bde3, jan 17 2021. [Online; accessed 2023-01-02].

[19] Simplilearn Simplilearn. Random Forest Algorithm. https://www.simplilearn.com/tutorials/machine-learning-tutorial/random-forest-algorithm. [Online; accessed 2023-01-02].

7 Appendices

This is the link for the python code[Final dissertation code (1).ipynb]. Here, we are importing libraries which contain functions.

```
In [1]: import pandas as pd
        import numpy as np
        import matplotlib.pyplot as plt
        import seaborn as sns
        import plotly.express as px
        from sklearn import preprocessing
        from sklearn.model_selection import train_test_split
        from sklearn import ensemble
        from sklearn.multioutput import MultiOutputClassifier
        from sklearn import multioutput
```

Reading the dataset1 and printing its shape:
First, we must load the dataset and read the excel file into one variable. Then summarize the dataset. This dataset contains 324 rows of data and 161 features or columns.

```
In [2]: df1 = pd.read_excel('Sheet1.xls')

        print(df1.shape)
        print(df1.head(5))
```

Reading the dataset2

This is the second dataset which is staff survey data. The above dataset is students' survey data. Load dataset 2 and read the second dataset into the second variable. Then summarize the dataset.

```
In [2]: df2 = pd.read_excel('Sheet2.xls')
        print(df2.shape)
        print(df2.head(5))
```

merging two datasets into one dataset

Now we have two datasets, merge these two datasets into a single dataset and reset the index. From this dataset, we will do further processes.

```
df3=pd.concat([df1,df2]).reset_index(drop=True)
print(df3.shape)
print(df3.head(5))
```

Basic information about dataset

Below are two lines of syntax explaining how the dataset would be in statistics. The first line of code is giving basic information about the dataset i.e., it contains 161 columns, 755 rows and the datatype of 161 features. There are two types of data in our dataset. They are 123 float types and 38 object types. The second line code describes the statistics of these float-type features.

```
#Basic information about dataset

df3.info()

#Describe the data

df3.describe()
```

dropping unwanted columns and renaming the column names

Here, we are removing unwanted columns which not useful for our prediction. The columns are too large so better to rename the columns.

```
df3 = df3.drop(df3.columns[[0,3,7,160]],axis = 1)
df3.rename(columns = {'2. Are you staff or student?':'2.member', '3. Are you:':'3.gender', '4. What is your term time or residential post code? This question is op
df3.rename(columns = {'12. If you have changed the mode of transport you use on your daily journey to the University over the past two years, please can you indica
# df3.head(5)
```

Checking duplicate and null values

In the dataset, first, we must check duplicate and null values. Our dataset has no duplicate values. But we have null values in a few columns. So, we must handle null values.

```
#Find the duplicates
print(df3.duplicated().sum())

#Find null values
print(df3.isnull().sum())
```

Handling missing values

This is the main process in data cleaning. We must deal with the missing values by using several methods. Let's get into the process.

Filling missing values using fillna() method

Here we are using fillna() method to fill in the missing values in these columns. The missing values are placed by 0.

```
df3[['7.1.a. Bus - Monday','7.1.b. Bus - Tuesday','7.1.c. Bus - Wednesday','7.1.d. Bus - Thursday','7.1.e. Bus - Friday','7.5.f. Bus - Everyday']] = df3[['7.1.a. B
df3[['7.2.a. Rail - Monday','7.2.b. Rail - Tuesday','7.2.c. Rail - Wednesday','7.2.d. Rail - Thursday','7.2.e. Rail - Friday','7.2.f. Rail - Everyday']] = df3[['7.
df3[['7.3.a. Metro - Monday','7.3.b. Metro - Tuesday','7.3.c. Metro - Wednesday','7.3.d. Metro - Thursday','7.3.e. Metro - Friday','7.3.f. Metro - Everyday']] = df
df3[['7.4.a. Walk - Monday','7.4.b. Walk - Tuesday','7.4.c. Walk - Wednesday','7.4.d. Walk - Thursday','7.4.e. Walk - Friday','7.4.f. Walk - Everyday']] = df3[['7.
df3[['7.5.a. Cycle - Monday','7.5.b. Cycle - Tuesday','7.5.c. Cycle - Wednesday','7.5.d. Cycle - Thursday','7.5.e. Cycle - Friday','7.5.f. Cycle - Everyday']] = df
df3[['7.6.a. Drive all the way (lone driver) - Monday','7.6.b. Drive all the way (lone driver) - Tuesday','7.6.c. Drive all the way (lone driver) - Wednesday','7.6
df3[['7.7.a. Car passenger (dropped off) - Monday','7.7.b. Car passenger (dropped off) - Tuesday','7.7.c. Car passenger (dropped off) - Wednesday','7.7.d. Car pass
df3[['7.8.a. Car share (with someone at the University) - Monday','7.8.b. Car share (with someone at the University) - Tuesday','7.8.c. Car share (with someone at
df3[['7.9.a. Park and Ride (E.g. car to train/bus then public transport) - Monday','7.9.b. Park and Ride (E.g. car to train/bus then public transport) - Tuesday','
df3[['7.10.a. Other (please specify) - Monday','7.10.b. Other (please specify) - Tuesday','7.10.c. Other (please specify) - Wednesday','7.10.d. Other (please speci

df3[['8.1.a. Bus - Monday','8.1.b. Bus - Tuesday','8.1.c. Bus - Wednesday','8.1.d. Bus - Thursday','8.1.e. Bus - Friday','8.1.f. Bus - Everyday']] = df3[['8.1.a. B
df3[['8.2.a. Rail - Monday','8.2.b. Rail - Tuesday','8.2.c. Rail - Wednesday','8.2.d. Rail - Thursday','8.2.e. Rail - Friday','8.2.f. Rail - Everyday']] = df3[['8.
df3[['8.3.a. Metro - Monday','8.3.b. Metro - Tuesday','8.3.c. Metro - Wednesday','8.3.d. Metro - Thursday','8.3.e. Metro - Friday','8.3.f. Metro - Everyday']] = df
df3[['8.4.a. Walk - Monday','8.4.b. Walk - Tuesday','8.4.c. Walk - Wednesday','8.4.d. Walk - Thursday','8.4.e. Walk - Friday','8.4.f. Walk - Everyday']] = df3[['8.
df3[['8.5.a. Cycle - Monday','8.5.b. Cycle - Tuesday','8.5.c. Cycle - Wednesday','8.5.d. Cycle - Thursday','8.5.e. Cycle - Friday','8.5.f. Cycle - Everyday']] = df
df3[['8.6.a. Drive all the way (lone driver) - Monday','8.6.b. Drive all the way (lone driver) - Tuesday','8.6.c. Drive all the way (lone driver) - Wednesday','8.6
df3[['8.7.a. Car passenger (dropped off) - Monday','8.7.b. Car passenger (dropped off) - Tuesday','8.7.c. Car passenger (dropped off) - Wednesday','8.7.d. Car pass
df3[['8.8.a. Car share (with someone at the University) - Monday','8.8.b. Car share (with someone at the University) - Tuesday','8.8.c. Car share (with someone at
df3[['8.9.a. Park and Ride (E.g. car to train/bus then public transport) - Monday','8.9.b. Park and Ride (E.g. car to train/bus then public transport) - Tuesday','
df3[['8.10.a. Other (please specify) - Monday','8.10.b. Other (please specify) - Tuesday','8.10.c. Other (please specify) - Wednesday','8.10.d. Other (please speci

df3.head(5)
```

Here filling one column values by adding multiple column values

We have so many columns in our dataset. In our dataset, they gave which day people will come by bus, train, car, metro etc. Here, we can reduce columns by summing multiple columns and placing values in every day named columns which means how many days people are coming by these transport modes in these columns(count).

```
df3['7.5.f. Bus - Everyday'] = (df3[['7.1.a. Bus - Monday','7.1.b. Bus - Tuesday','7.1.c. Bus - Wednesday','7.1.d. Bus - Thursday','7.1.e. Bus - Friday','7.5.f. Bu
df3['7.2.f. Rail - Everyday'] = (df3[['7.2.a. Rail - Monday','7.2.b. Rail - Tuesday','7.2.c. Rail - Wednesday','7.2.d. Rail - Thursday','7.2.e. Rail - Friday','7.2
df3['7.3.f. Metro - Everyday'] = (df3[['7.3.a. Metro - Monday','7.3.b. Metro - Tuesday','7.3.c. Metro - Wednesday','7.3.d. Metro - Thursday','7.3.e. Metro - Friday
df3['7.4.f. Walk - Everyday'] = (df3[['7.4.a. Walk - Monday','7.4.b. Walk - Tuesday','7.4.c. Walk - Wednesday','7.4.d. Walk - Thursday','7.4.e. Walk - Friday','7.4
df3['7.5.f. Cycle - Everyday'] = (df3[['7.5.a. Cycle - Monday','7.5.b. Cycle - Tuesday','7.5.c. Cycle - Wednesday','7.5.d. Cycle - Thursday','7.5.e. Cycle - Friday
df3['7.6.f. Drive all the way (lone driver) - Everyday'] = (df3[['7.6.a. Drive all the way (lone driver) - Monday','7.6.b. Drive all the way (lone driver) - Tuesda
df3['7.7.f. Car passenger (dropped off) - Everyday'] = (df3[['7.7.a. Car passenger (dropped off) - Monday','7.7.b. Car passenger (dropped off) - Tuesday','7.7.c. C
df3['7.8.f. Car share (with someone at the University) - Everyday'] = (df3[['7.8.a. Car share (with someone at the University) - Monday','7.8.b. Car share (with so
df3['7.9.f. Park and Ride (E.g. car to train/bus then public transport) - Everyday'] = (df3[['7.9.a. Park and Ride (E.g. car to train/bus then public transport) -
df3['7.50.f. Other (please specify) - Everyday'] = (df3[['7.10.a. Other (please specify) - Monday','7.10.b. Other (please specify) - Tuesday','7.10.c. Other (pleas

df3['8.1.f. Bus - Everyday'] = (df3[['8.1.a. Bus - Monday','8.1.b. Bus - Tuesday','8.1.c. Bus - Wednesday','8.1.d. Bus - Thursday','8.1.e. Bus - Friday','8.1.f. Bu
df3['8.2.f. Rail - Everyday'] = (df3[['8.2.a. Rail - Monday','8.2.b. Rail - Tuesday','8.2.c. Rail - Wednesday','8.2.d. Rail - Thursday','8.2.e. Rail - Friday','8.2
df3['8.3.f. Metro - Everyday'] = (df3[['8.3.a. Metro - Monday','8.3.b. Metro - Tuesday','8.3.c. Metro - Wednesday','8.3.d. Metro - Thursday','8.3.e. Metro - Friday
df3['8.4.f. Walk - Everyday'] = (df3[['8.4.a. Walk - Monday','8.4.b. Walk - Tuesday','8.4.c. Walk - Wednesday','8.4.d. Walk - Thursday','8.4.e. Walk - Friday','8.4
df3['8.5.f. Cycle - Everyday'] = (df3[['8.5.a. Cycle - Monday','8.5.b. Cycle - Tuesday','8.5.c. Cycle - Wednesday','8.5.d. Cycle - Thursday','8.5.e. Cycle - Friday
df3['8.6.f. Drive all the way (lone driver) - Everyday'] = (df3[['8.6.a. Drive all the way (lone driver) - Monday','8.6.b. Drive all the way (lone driver) - Tuesda
df3['8.7.f. Car passenger (dropped off) - Everyday'] = (df3[['8.7.a. Car passenger (dropped off) - Monday','8.7.b. Car passenger (dropped off) - Tuesday','8.7.c. C
df3['8.8.f. Car share (with someone at the University) - Everyday'] = (df3[['8.8.a. Car share (with someone at the University) - Monday','8.8.b. Car share (with so
df3['8.9.f. Park and Ride (E.g. car to train/bus then public transport) - Everyday'] = (df3[['8.9.a. Park and Ride (E.g. car to train/bus then public transport) -
df3['8.10.f. Other (please specify) - Everyday'] = (df3[['8.10.a. Other (please specify) - Monday','8.10.b. Other (please specify) - Tuesday','8.10.c. Other (pleas
```

Removing unwanted columns

Above we filled few columns by adding multiple columns. So, we are keeping those columns and removing multiple columns.

```
df3 = df3.drop(df3.columns[[5,6,7,8,9,11,12,13,14,15,17,18,19,20,21,23,24,25,26,27,29,30,31,32,33,35,36,37,38,39,41,42,43,44,45,47,48,49,50,51,53,54,55,56,57,59,60
df3.head(5)
```

Removing columns based on percentage of missing values

We are removing columns based on percentage of missing values. Those columns are having greater than 80% of missing values, then those columns will be deleted.

```
m_null_percentage = df3.isnull().mean()*100
coln_to_drop = m_null_percentage[m_null_percentage>80].keys()
# (8.a,10.a,11.a,12.a,18.a,27,33)
df3 = df3.drop(coln_to_drop, axis=1)
df3.shape
```

Describe function describes the statistics of these numerical features.

```
df3.describe()
```

Filling missing values using value_counts method

This is another method is used to fill in the missing values. This column('16. At what time do you normally depart from campus?') has 8 missing values. So, we can use the value_counts method to fill these values.

```
df3 = df3.drop('4.postcode', axis=1)
df3['16. At what time do you normally depart from campus?'].fillna(df3['16. At what time do you normally depart from campus?'].value_counts().index[0], inplace = T
```

Data Visualization

percentages of students that come by travel mode to university

Here, the percentages of students that come by bus, rail, metro, walk, cycle, drive all the way, car share, park and ride (E.g., car to train/bus then public transport), other etc are calculated.

```
# general mode to university by Student
# https://www.statology.org/pandas-sum-column-with-condition/
Bus = df3.loc[df3['2.member'] == 'Student', '7.5.f. Bus - Everyday'].sum()
Rail = df3.loc[df3['2.member'] == 'Student', '7.2.f. Rail - Everyday'].sum()
Metro = df3.loc[df3['2.member'] == 'Student', '7.3.f. Metro - Everyday'].sum()
Walk = df3.loc[df3['2.member'] == 'Student', '7.4.f. Walk - Everyday'].sum()
Cycle = df3.loc[df3['2.member'] == 'Student', '7.5.f. Cycle - Everyday'].sum()
Drive_all_the_way_lone_driver = df3.loc[df3['2.member'] == 'Student', '7.6.f. Drive all the way (lone driver) - Everyday'].sum()
Car_passenger_dropped_off = df3.loc[df3['2.member'] == 'Student', '7.7.f. Car passenger (dropped off) - Everyday'].sum()
Car_share_with_someone_at_the_University  = df3.loc[df3['2.member'] == 'Student', '7.8.f. Car share (with someone at the University) - Everyday'].sum()
Park_and_Ride_car_to_train_bus_then_public_transport = df3.loc[df3['2.member'] == 'Student', '7.9.f. Park and Ride (E.g. car to train/bus then public transport) -
Other_please_specify =  df3.loc[df3['2.member'] == 'Student', '7.50.f. Other (please specify) - Everyday'].sum()

total = Bus+ Rail+ Metro+ Walk+ Cycle+ Drive_all_the_way_lone_driver+ Car_passenger_dropped_off+ Car_share_with_someone_at_the_University+ Park_and_Ride_car_to_tra


Bus_percentage = Bus/total*100
Rail_percentage = Rail/total*100
Metro_percentage = Metro/total*100
Walk_percentage = Walk/total*100
Cycle_percentage = Cycle/total*100
Drive_all_the_way_lone_driver_percentage = Drive_all_the_way_lone_driver/total*100
Car_passenger_dropped_off_percentage = Car_passenger_dropped_off/total*100
Car_share_with_someone_at_the_University_percentage = Car_share_with_someone_at_the_University/total*100
Park_and_Ride_car_to_train_bus_then_public_transport_percentage = Park_and_Ride_car_to_train_bus_then_public_transport/total*100
Other_please_specify_percentage = Other_please_specify/total*100
print('Bus_percentage =', Bus_percentage)
print('Rail_percentage =',Rail_percentage)
print('Metro_percentage =',Metro_percentage)
print('Walk_percentage =',Walk_percentage)
```

```
print('Cycle_percentage =',Cycle_percentage)
print('Drive_all_the_way_lone_driver_percentage =',Drive_all_the_way_lone_driver_percentage)
print('Car_passenger_dropped_off_percentage =',Car_passenger_dropped_off_percentage)
print('Car_share_with_someone_at_the_University_percentage =',Car_share_with_someone_at_the_University_percentage)
print('Park_and_Ride_car_to_train_bus_then_public_transport_percentage =',Park_and_Ride_car_to_train_bus_then_public_transport_percentage)
print('Other_please_specify_percentage =',Other_please_specify_percentage)
```

Below is the distribution of transport modes by students travel to the university.

```
ag=pd.DataFrame({'Travel mode':['Bus','Rail','Metro','Walk','Cycle','Drive_all_the_way_lone_driver','Car_passenger_dropped_off','Car_share_with_someone_at_the_Univ

fig=px.bar(ag,x='Travel mode',y='Percentage',color='Travel mode',template='simple_white',text='Percentage',title='what are the percentages of how students get into
fig.update_traces(marker=dict(line=dict(color='#000000', width=1)))
```

percentages of staff that come by travel mode to university

Here, the percentages of staff that come by bus, rail, metro, walk, cycle, drive all the way, car share, park and ride (E.g., car to train/bus then public transport), other etc are calculated.

```
# general mode to university by Staff
Bus_staff = df3.loc[df3['2.member'] == 'Staff', '7.5.f. Bus - Everyday'].sum()
Rail_staff = df3.loc[df3['2.member'] == 'Staff', '7.2.f. Rail - Everyday'].sum()
Metro_staff = df3.loc[df3['2.member'] == 'Staff', '7.3.f. Metro - Everyday'].sum()
Walk_staff = df3.loc[df3['2.member'] == 'Staff', '7.4.f. Walk - Everyday'].sum()
Cycle_staff = df3.loc[df3['2.member'] == 'Staff', '7.5.f. Cycle - Everyday'].sum()
Drive_all_the_way_lone_driver_staff = df3.loc[df3['2.member'] == 'Staff', '7.6.f. Drive all the way (lone driver) - Everyday'].sum()
Car_passenger_dropped_off_staff = df3.loc[df3['2.member'] == 'Staff', '7.7.f. Car passenger (dropped off) - Everyday'].sum()
Car_share_with_someone_at_the_University_staff  = df3.loc[df3['2.member'] == 'Staff', '7.8.f. Car share (with someone at the University) - Everyday'].sum()
Park_and_Ride_car_to_train_bus_then_public_transport_staff = df3.loc[df3['2.member'] == 'Staff', '7.9.f. Park and Ride (E.g. car to train/bus then public transport
Other_please_specify_staff =  df3.loc[df3['2.member'] == 'Staff', '7.50.f. Other (please specify) - Everyday'].sum()

total_staff = Bus_staff+ Rail_staff+ Metro_staff+ Walk_staff+ Cycle_staff+ Drive_all_the_way_lone_driver_staff+ Car_passenger_dropped_off_staff+ Car_share_with_som


Bus_percentage_staff = Bus_staff/total_staff*100
Rail_percentage_staff = Rail_staff/total_staff*100
Metro_percentage_staff = Metro_staff/total_staff*100
```

```
Metro_percentage_staff = Metro_staff/total_staff*100
Walk_percentage_staff = Walk_staff/total_staff*100
Cycle_percentage_staff = Cycle_staff/total_staff*100
Drive_all_the_way_lone_driver_percentage_staff = Drive_all_the_way_lone_driver_staff/total_staff*100
Car_passenger_dropped_off_percentage_staff = Car_passenger_dropped_off_staff/total_staff*100
Car_share_with_someone_at_the_University_percentage_staff = Car_share_with_someone_at_the_University_staff/total_staff*100
Park_and_Ride_car_to_train_bus_then_public_transport_percentage_staff = Park_and_Ride_car_to_train_bus_then_public_transport_staff/total_staff*100
Other_please_specify_percentage_staff = Other_please_specify_staff/total_staff*100
print('Bus_percentage_staff =',Bus_percentage_staff)
print('Rail_percentage_staff =',Rail_percentage_staff)
print('Metro_percentage_staff  =',Metro_percentage_staff)
print('Walk_percentage_staff =',Walk_percentage_staff)
print('Cycle_percentage_staff =',Cycle_percentage_staff)
print('Drive_all_the_way_lone_driver_percentage_staff =',Drive_all_the_way_lone_driver_percentage_staff)
print('Car_passenger_dropped_off_percentage_staff =',Car_passenger_dropped_off_percentage_staff)
print('Car_share_with_someone_at_the_University_percentage_staff =',Car_share_with_someone_at_the_University_percentage_staff)
print('Park_and_Ride_car_to_train_bus_then_public_transport_percentage_staff =',Park_and_Ride_car_to_train_bus_then_public_transport_percentage_staff)
print('Other_please_specify_percentage_staff =',Other_please_specify_percentage_staff)
```

Below is the distribution of transport modes by staff travel to the university.

```
]:
ag=pd.DataFrame({'Travel mode':['Bus','Rail','Metro','Walk','Cycle','Drive_all_the_way_lone_driver','Car_passenger_dropped_off','Car_share_with_someone_at_the_Univ

fig=px.bar(ag,x='Travel mode',y='Percentage',color='Travel mode',template='simple_white',text='Percentage',title='what are the percentages of how staff get into ca
fig.update_traces(marker=dict(line=dict(color='#000000', width=1)))
```

percentages of students that come by travel mode from university

Here, the percentages of students that come by bus, rail, metro, walk, cycle, drive all the way, car share, park and ride (E.g., car to train/bus then public transport), other etc are calculated.

```
]:
# general mode travel FROM Aston University
Bus_univ = df3.loc[df3['2.member'] == 'Student', '8.1.f. Bus - Everyday'].sum()
Rail_univ = df3.loc[df3['2.member'] == 'Student', '8.2.f. Rail - Everyday'].sum()
Metro_univ = df3.loc[df3['2.member'] == 'Student', '8.3.f. Metro - Everyday'].sum()
Walk_univ = df3.loc[df3['2.member'] == 'Student', '8.4.f. Walk - Everyday'].sum()
Cycle_univ = df3.loc[df3['2.member'] == 'Student', '8.5.f. Cycle - Everyday'].sum()
```

```
Drive_all_the_way_lone_driver_univ = df3.loc[df3['2.member'] == 'Student', '8.6.f. Drive all the way (lone driver) - Everyday'].sum()
Car_passenger_dropped_off_univ = df3.loc[df3['2.member'] == 'Student', '8.7.f. Car passenger (dropped off) - Everyday'].sum()
Car_share_with_someone_at_the_University_univ  = df3.loc[df3['2.member'] == 'Student', '8.8.f. Car share (with someone at the University) - Everyday'].sum()
Park_and_Ride_car_to_train_bus_then_public_transport_univ = df3.loc[df3['2.member'] == 'Student', '8.9.f. Park and Ride (E.g. car to train/bus then public transpor
Other_please_specify_univ =  df3.loc[df3['2.member'] == 'Student', '8.10.f. Other (please specify) - Everyday'].sum()


total = Bus_univ+ Rail_univ+ Metro_univ+ Walk_univ+ Cycle_univ+ Drive_all_the_way_lone_driver_univ+ Car_passenger_dropped_off_univ+ Car_share_with_someone_at_the_U


Bus_percentage_univ = Bus_univ/total*100
Rail_percentage_univ = Rail_univ/total*100
Metro_percentage_univ = Metro_univ/total*100
Walk_percentage_univ = Walk_univ/total*100
Cycle_percentage_univ = Cycle_univ/total*100
Drive_all_the_way_lone_driver_percentage_univ = Drive_all_the_way_lone_driver_univ/total*100
Car_passenger_dropped_off_percentage_univ = Car_passenger_dropped_off_univ/total*100
Car_share_with_someone_at_the_University_percentage_univ = Car_share_with_someone_at_the_University_univ/total*100
Park_and_Ride_car_to_train_bus_then_public_transport_percentage_univ = Park_and_Ride_car_to_train_bus_then_public_transport_univ/total*100
Other_please_specify_percentage_univ = Other_please_specify_univ/total*100
print('Bus_percentage_univ =',Bus_percentage_univ)
print('Rail_percentage_univ =',Rail_percentage_univ)
print('Metro_percentage_univ =',Metro_percentage_univ)
print('Walk_percentage_univ =',Walk_percentage_univ)
print('Cycle_percentage_univ =',Cycle_percentage_univ)
print('Drive_all_the_way_lone_driver_percentage_univ =',Drive_all_the_way_lone_driver_percentage_univ)
print('Car_passenger_dropped_off_percentage_univ =',Car_passenger_dropped_off_percentage_univ)
print('Car_share_with_someone_at_the_University_percentage_univ =',Car_share_with_someone_at_the_University_percentage_univ)
print('Park_and_Ride_car_to_train_bus_then_public_transport_percentage_univ =',Park_and_Ride_car_to_train_bus_then_public_transport_percentage_univ)
print('Other_please_specify_percentage_univ =',Other_please_specify_percentage_univ)
```

Below is the distribution of transport modes by students travel from the university.

```
]:
ag=pd.DataFrame({'Travel mode':['Bus','Rail','Metro','Walk','Cycle','Drive_all_the_way_lone_driver','Car_passenger_dropped_off','Car_share_with_someone_at_the_Univ

fig=px.bar(ag,x='Travel mode',y='Percentage',color='Travel mode',template='simple_white',text='Percentage',title='what are the percentages of how students go home
fig.update_traces(marker=dict(line=dict(color='#000000', width=1)))
```

```
fig=px.bar(ag,x='Travel mode',y='Percentage',color='Travel mode',template='simple_white',text='Percentage',title='what are the percentages of how students go home
fig.update_traces(marker=dict(line=dict(color='#000000', width=1)))
```

percentages of staff that come by travel mode from university

Here, the percentages of staff that come by bus, rail, metro, walk, cycle, drive all the way, car share, park and ride (E.g., car to train/bus then public transport), other etc are calculated.

```
:
# general mode travel FROM Aston University
Bus_univ = df3.loc[df3['2.member'] == 'Staff', '8.1.f. Bus - Everyday'].sum()
Rail_univ = df3.loc[df3['2.member'] == 'Staff', '8.2.f. Rail - Everyday'].sum()
Metro_univ = df3.loc[df3['2.member'] == 'Staff', '8.3.f. Metro - Everyday'].sum()
Walk_univ = df3.loc[df3['2.member'] == 'Staff', '8.4.f. Walk - Everyday'].sum()
Cycle_univ = df3.loc[df3['2.member'] == 'Staff', '8.5.f. Cycle - Everyday'].sum()
Drive_all_the_way_lone_driver_univ = df3.loc[df3['2.member'] == 'Staff', '8.6.f. Drive all the way (lone driver) - Everyday'].sum()
Car_passenger_dropped_off_univ = df3.loc[df3['2.member'] == 'Staff', '8.7.f. Car passenger (dropped off) - Everyday'].sum()
Car_share_with_someone_at_the_University_univ  = df3.loc[df3['2.member'] == 'Staff', '8.8.f. Car share (with someone at the University) - Everyday'].sum()
Park_and_Ride_car_to_train_bus_then_public_transport_univ = df3.loc[df3['2.member'] == 'Staff', '8.9.f. Park and Ride (E.g. car to train/bus then public transport)
Other_please_specify_univ =  df3.loc[df3['2.member'] == 'Staff', '8.10.f. Other (please specify) - Everyday'].sum()
```

```
total = Bus_univ+ Rail_univ+ Metro_univ+ Walk_univ+ Cycle_univ+ Drive_all_the_way_lone_driver_univ+ Car_passenger_dropped_off_univ+ Car_share_with_someone_at_the_U


Bus_percentage_univ = Bus_univ/total*100
Rail_percentage_univ = Rail_univ/total*100
Metro_percentage_univ = Metro_univ/total*100
Walk_percentage_univ = Walk_univ/total*100
Cycle_percentage_univ = Cycle_univ/total*100
Drive_all_the_way_lone_driver_percentage_univ = Drive_all_the_way_lone_driver_univ/total*100
Car_passenger_dropped_off_percentage_univ = Car_passenger_dropped_off_univ/total*100
Car_share_with_someone_at_the_University_percentage_univ = Car_share_with_someone_at_the_University_univ/total*100
Park_and_Ride_car_to_train_bus_then_public_transport_percentage_univ = Park_and_Ride_car_to_train_bus_then_public_transport_univ/total*100
Other_please_specify_percentage_univ = Other_please_specify_univ/total*100

print('Bus_percentage_univ =',Bus_percentage_univ)
print('Rail_percentage_univ =',Rail_percentage_univ)
```

```
print('Walk_percentage_univ =',Walk_percentage_univ)
print('Cycle_percentage_univ =',Cycle_percentage_univ)
print('Drive_all_the_way_lone_driver_percentage_univ =',Drive_all_the_way_lone_driver_percentage_univ)
print('Car_passenger_dropped_off_percentage_univ =',Car_passenger_dropped_off_percentage_univ)
print('Car_share_with_someone_at_the_University_percentage_univ =',Car_share_with_someone_at_the_University_percentage_univ)
print('Park_and_Ride_car_to_train_bus_then_public_transport_percentage_univ =',Park_and_Ride_car_to_train_bus_then_public_transport_percentage_univ)
print('Other_please_specify_percentage_univ =',Other_please_specify_percentage_univ)
```

Below is the distribution of transport modes by staff travel from the university.

```
]:
ag=pd.DataFrame({'Travel mode':['Bus','Rail','Metro','Walk','Cycle','Drive_all_the_way_lone_driver','Car_passenger_dropped_off','Car_share_with_someone_at_the_Univ

fig=px.bar(ag,x='Travel mode',y='Percentage',color='Travel mode',template='simple_white',text='Percentage',title='what are the percentages of how staff go home fro
fig.update_traces(marker=dict(line=dict(color='#000000', width=1)))
```

The below plot represents, the number of students and staff who respond to this survey i.e.,324 students and 431 staffin this dataset.

The below plot represents, the number of students and staff who respond to this survey i.e.,324 students and 431 staffin this dataset.

```
]:
print(df3['2.member'].value_counts())
sns.countplot(x='2.member',hue='2.member',data=df3)
plt.xlabel('Member')
plt.ylabel('count')
plt.title('plot')
plt.show()
```

plot which shows the relation between member(students and staff) and gender.

```
]:
df3.value_counts(["2.member", "3.gender"])
```

```
]:
sns.countplot(x ='2.member', hue = "3.gender", data = df3)
plt.title('plot which shows relation between (students & staff) and gender')
# Show the plot
plt.show()
```

plot which shows the relation between member(students and staff) and age of them.

```
:
df3.value_counts(["2.member", "5.age"])
```

```
:
sns.countplot(x ='2.member', hue = '5.age', data = df3)
plt.title('plot which shows relation between (students & staff) and agegroup')
# Show the plot
plt.show()
```

plot which shows the relation between member(students and staff) and how far from home to the workplace.

```
:
fig, ax = plt.subplots(figsize=(8, 4))
sns.countplot(x ='2.member', hue = '6.how far from home to workplace', data = df3)
plt.title('plot which shows relation between (students & staff) and distance from the house to the workplace')
# Show the plot
plt.show()
```

This plot is representing the relation between member(students and staff) and people who change their mode of travel between summer and winter. So, based on this we can know how many people are changing their travel mode.

```
:
fig, ax = plt.subplots(figsize=(8, 4))
sns.countplot(x ='2.member', hue = '9.mode of travel change based on season', data = df3)
plt.title('plot which shows relation between (students & staff) and mode of travel changed based on season')
# Show the plot
plt.show()
```

Below pie chart represents 'why people travel to the university in the way they do?'.

```
:
fig = plt.figure(figsize =(4, 4))
values = df3['10. Why do you normally travel to the University in the way you do?'].value_counts()
labels = df3['10. Why do you normally travel to the University in the way you do?'].unique().tolist()
# fig.show()
print(values)
```

```
print(values)
px.pie(df3, values = values, labels = labels, title = '10. Why do you normally travel to the University in the way you do?')
```

This plot is representing the relation between member(students and staff) and how often do they travel to the University.

```
df3.value_counts(["2.member", "14. How often do you travel to the University?"])
```

```
fig, ax = plt.subplots(figsize=(8, 4))
sns.countplot(x ='2.member', hue = '14. How often do you travel to the University?', data = df3)
plt.title('plot which shows relation between (students & staff) and How often do you travel to the University')
# Show the plot
plt.show()
```

Here we are encoding the data using a label encoder. The Label Encoder is used to convert categorical feature values into numeric values. Then we use those values to predict missing values in a few columns.

```
lab2_encoder = preprocessing.LabelEncoder()
lab3_encoder = preprocessing.LabelEncoder()
lab5_encoder = preprocessing.LabelEncoder()
lab6_encoder = preprocessing.LabelEncoder()
lab9_encoder = preprocessing.LabelEncoder()
lab10_encoder = preprocessing.LabelEncoder()
lab11_encoder = preprocessing.LabelEncoder()
lab12_encoder = preprocessing.LabelEncoder()
lab13_encoder = preprocessing.LabelEncoder()
lab14_encoder = preprocessing.LabelEncoder()
lab15_encoder = preprocessing.LabelEncoder()
lab16_encoder = preprocessing.LabelEncoder()
lab17_encoder = preprocessing.LabelEncoder()
lab18_encoder = preprocessing.LabelEncoder()
lab19_encoder = preprocessing.LabelEncoder()
lab20_encoder = preprocessing.LabelEncoder()
lab21_encoder = preprocessing.LabelEncoder()
lab22_encoder = preprocessing.LabelEncoder()
lab23_encoder = preprocessing.LabelEncoder()
lab24_encoder = preprocessing.LabelEncoder()
```

```
df3['2.member'] = lab2_encoder.fit_transform(df3['2.member'])
df3['3.gender'] = lab3_encoder.fit_transform(df3['3.gender'])
df3['5.age'] = lab5_encoder.fit_transform(df3['5.age'])
df3['6.how far from home to workplace'] = lab6_encoder.fit_transform(df3['6.how far from home to workplace'])
df3['9.mode of travel change based on season'] = lab9_encoder.fit_transform(df3['9.mode of travel change based on season'])
df3['10. Why do you normally travel to the University in the way you do?'] = lab10_encoder.fit_transform(df3['10. Why do you normally travel to the University in t
df3['11.transport mode preference to work'] = lab11_encoder.fit_transform(df3['11.transport mode preference to work'])
df3['13. Have you ever booked a University pool car - the electric BMW i3 vehicle (Staff only)?'] = lab13_encoder.fit_transform(df3['13. Have you ever booked a Uni
df3['14. How often do you travel to the University?'] = lab14_encoder.fit_transform(df3['14. How often do you travel to the University?'])
df3['15. At what time do you normally arrive at campus?'] = lab15_encoder.fit_transform(df3['15. At what time do you normally arrive at campus?'])
df3['16. At what time do you normally depart from campus?'] = lab16_encoder.fit_transform(df3['16. At what time do you normally depart from campus?'])
df3['17.journey time to the university'] = lab17_encoder.fit_transform(df3['17.journey time to the university'])
df3['24. Do you own or have access to a bicycle?'] = lab24_encoder.fit_transform(df3['24. Do you own or have access to a bicycle?'])
df3['25.do you have a secure bicycle parking in your residence'] = lab25_encoder.fit_transform(df3['25.do you have a secure bicycle parking in your residence'])
df3['26. Do you use the cycle parking available on-site at the University?'] = lab26_encoder.fit_transform(df3['26. Do you use the cycle parking available on-site
df3['28. What barriers do you have to accessing the University by public transport?'] = lab28_encoder.fit_transform(df3['28. What barriers do you have to accessing
df3['29. What barriers do you have to accessing the University by foot?'] = lab29_encoder.fit_transform(df3['29. What barriers do you have to accessing the Univers
df3['30. What barriers do you have to accessing the University by bike?'] = lab30_encoder.fit_transform(df3['30. What barriers do you have to accessing the Univers
df3['32. Which of the following would encourage you to travel more by public transport or improvements'] = lab30_encoder.fit_transform(df3['32. Which of the follow
```

```
df3_12 = df3[['12.reasons for the change of mode of transport used on your daily journey to the university over the past 2 years']]
df3_18 = df3[['18.need car for other purposes']]
df3_19 = df3[['19. If you travel by car to the University, where do you normally park?']]
df3_20 = df3[['20. If you park the car within the campus, have you any difficulty finding a parking space?']]
df3_21 = df3[['21. What would you do if car parking was unavailable at the University?']]
df3_22 = df3[['22. Are you aware of the Aston University car share scheme?']]
df3_23 = df3[['23. Which of the following would encourage you to car share either as a driver or passenger or improvements']]
df3_31 = df3[['31. Which of the following would encourage you to travel here more by bicycle or improvements']]
```

here, we are encoding our data except notnull values.

```
df3_12 = df3_12.apply(lambda series:pd.Series(lab12_encoder.fit_transform(series[series.notnull()]),
                                              index=series[series.notnull()].index))
df3_18 = df3_18.apply(lambda series:pd.Series(lab18_encoder.fit_transform(series[series.notnull()]),
                                              index=series[series.notnull()].index))
df3_19 = df3_19.apply(lambda series:pd.Series(lab19_encoder.fit_transform(series[series.notnull()]),
                                              index=series[series.notnull()].index))
df3_20 = df3_20.apply(lambda series:pd.Series(lab20_encoder.fit_transform(series[series.notnull()]),
                                              index=series[series.notnull()].index))
df3_21 = df3_21.apply(lambda series:pd.Series(lab21_encoder.fit_transform(series[series.notnull()]),
                                              index=series[series.notnull()].index))
df3_22 = df3_22.apply(lambda series:pd.Series(lab22_encoder.fit_transform(series[series.notnull()]),
                                              index=series[series.notnull()].index))
df3_23 = df3_23.apply(lambda series:pd.Series(lab23_encoder.fit_transform(series[series.notnull()]),
                                              index=series[series.notnull()].index))
df3_31 = df3_31.apply(lambda series:pd.Series(lab31_encoder.fit_transform(series[series.notnull()]),
                                              index=series[series.notnull()].index))
```

```
df3[['12.reasons for the change of mode of transport used on your daily journey to the university over the past 2 years']] = df3_12
df3[['18.need car for other purposes']] = df3_18
df3[['19. If you travel by car to the University, where do you normally park?']] = df3_19
df3[['20. If you park the car within the campus, have you any difficulty finding a parking space?']] = df3_20
df3[['21. What would you do if car parking was unavailable at the University?']] = df3_21
df3[['22. Are you aware of the Aston University car share scheme?']] = df3_22
df3[['23. Which of the following would encourage you to car share either as a driver or passenger or improvements']] = df3_23
df3[['31. Which of the following would encourage you to travel here more by bicycle or improvements']] = df3_31
```

```
df3.head(5)
```

here we are predicting missing values of these 8 columns using the Random Forest model. First, we separated the null value dataset as testing and the nonnull value dataset as the training dataset.

```
# dropping nan value columns except predicting column
df3_12 = df3.drop(df3.columns[[33,34,35,36,37,38,45]],axis = 1)
df3_12

# Splitting dataset into two i.e, non-null as training dataset and null as testing dataset
# non-null as training dataset
df3_12_train = df3_12[df3_12['12.reasons for the change of mode of transport used on your daily journey to the university over the past 2 years'].notna()]
# print(df3_12_train.shape)
```

```
# null as testing dataset
df3_12_test = df3_12[df3_12['12.reasons for the change of mode of transport used on your daily journey to the university over the past 2 years'].isnull()]
df3_12_test.shape
# print(df3_12_train.isnull().sum())

x_train_12 = df3_12_train.drop('12.reasons for the change of mode of transport used on your daily journey to the university over the past 2 years',axis = 1)
x_train_12

y_train_12 = df3_12_train['12.reasons for the change of mode of transport used on your daily journey to the university over the past 2 years']
y_train_12
# print(x_train_12.isnull().sum())
# print('\n', y_train_12.isnull().sum())

from sklearn.ensemble import RandomForestClassifier
r_forest = RandomForestClassifier()
r_forest.fit(x_train_12, y_train_12)

# dataset excepts '18.need car for other purposes' feature with null values
x_test_12 = df3_12_test.drop('12.reasons for the change of mode of transport used on your daily journey to the university over the past 2 years',axis = 1)
x_test_12

y_pred_12 = r_forest.predict(x_test_12)
y_pred_12

#creating a new dataset to store prediction values and ID position
df_null_12 = pd.DataFrame(y_pred_12, columns=['prevision'])

#reset index on null dataset
nan_12 = df3_12_test.copy().reset_index()

#add column in the null dataset
df_prev_12 = pd.concat([nan_12, df_null_12], axis=1)

#set index
df_prev_12 = df_prev_12.set_index('index')

#fill the values
df3 = df3.fillna({'12.reasons for the change of mode of transport used on your daily journey to the university over the past 2 years':df_prev_12['prevision']}).cop
```

```
df3.head(5)
```

here we are predicting missing values of '18.need car for other purposes' using the Random Forest model. First, we separated the null value dataset as testing and the nonnull value dataset as the training dataset.

```
# dropping nan value columns except predicting column
df3_18 = df3.drop(df3.columns[[34,35,36,37,38,45]],axis = 1)
df3_18

# Splitting dataset into two i.e, non-null as training dataset and null as testing dataset
# non-null as training dataset
df3_18_train = df3_18[df3_18['18.need car for other purposes'].notna()]
df3_18_train.shape

# null as testing dataset
df3_18_test = df3_18[df3_18['18.need car for other purposes'].isnull()]
df3_18_test.shape
# print(df3_18_train.isnull().sum())

x_train_18 = df3_18_train.drop('18.need car for other purposes',axis = 1)
x_train_18.shape

y_train_18 = df3_18_train['18.need car for other purposes']
y_train_18

# print(x_train_18.isnull().sum())
# print('\n', y_train_18.isnull().sum())

from sklearn.ensemble import RandomForestClassifier
r_forest = RandomForestClassifier()
r_forest.fit(x_train_18, y_train_18)

# dataset excepts '18.need car for other purposes' feature with null values
x_test_18 = df3_18_test.drop('18.need car for other purposes',axis = 1)
x_test_18
```

```
y_pred_18 = r_forest.predict(x_test_18)
y_pred_18

#creating a new dataset to store prediction values and ID position
df_null_18 = pd.DataFrame(y_pred_18, columns=['prevision'])

#reset index on null dataset
nan_18 = df3_18_test.copy().reset_index()

#add column in the null dataset
df_prev_18 = pd.concat([nan_18, df_null_18], axis=1)

#set index
df_prev_18 = df_prev_18.set_index('index')

#fill the values
df3 = df3.fillna({'18.need car for other purposes':df_prev_18['prevision']}).copy()

# df3['18.need car for other purposes'] = lab_encoder.fit_transform(df3['18.need car for other purposes']).astype(int)
df3.head(5)
```

here we are predicting missing values of '19. If you travel by car to the University, where do you normally park?' using the Random Forest model. First, we separated the null value dataset as testing and the nonnull value dataset as the training dataset.

```
# dropping nan value columns except predicting column
df3_19 = df3.drop(df3.columns[[35,36,37,38,45]],axis = 1)
df3_19

# Splitting dataset into two i.e, non-null as training dataset and null as testing dataset
# non-null as training dataset
df3_19_train = df3_19[df3_19['19. If you travel by car to the University, where do you normally park?'].notna()]
df3_19_train.shape

# null as testing dataset
df3_19_test = df3_19[df3_19['19. If you travel by car to the University, where do you normally park?'].isnull()]
df3_19_test.shape
# print(df3_19_train.isnull().sum())
```

```
x_train_19 = df3_19_train.drop('19. If you travel by car to the University, where do you normally park?',axis = 1)
x_train_19

y_train_19 = df3_19_train['19. If you travel by car to the University, where do you normally park?']
y_train_19
# print(x_train_19.isnull().sum())
# print('\n', y_train_19.isnull().sum())

from sklearn.ensemble import RandomForestClassifier
r_forest = RandomForestClassifier()
r_forest.fit(x_train_19, y_train_19)

# dataset excepts '18.need car for other purposes' feature with null values
x_test_19 = df3_19_test.drop('19. If you travel by car to the University, where do you normally park?',axis = 1)
x_test_19

y_pred_19 = r_forest.predict(x_test_19)
y_pred_19

#creating a new dataset to store prediction values and ID position
df_null_19 = pd.DataFrame(y_pred_19, columns=['prevision'])

#reset index on null dataset
nan_19 = df3_19_test.copy().reset_index()

#add column in the null dataset
df_prev_19 = pd.concat([nan_19, df_null_19], axis=1)

#set index
df_prev_19 = df_prev_19.set_index('index')

#fill the values
df3 = df3.fillna({'19. If you travel by car to the University, where do you normally park?':df_prev_19['prevision']}).copy()

# df3['19. If you travel by car to the University, where do you normally park?'] = lab_encoder.fit_transform(df3['19. If you travel by car to the University, where
df3.head(5)
```

here we are predicting missing values of '20. If you park the car within the campus, have you any difficulty finding a parking space?' using the Random Forest model. First, we separated the

```
# dropping nan value columns except predicting column
df3_20 = df3.drop(df3.columns[[36,37,38,45]],axis = 1)
df3_20

# Splitting dataset into two i.e, non-null as training dataset and null as testing dataset
# non-null as training dataset
df3_20_train = df3_20[df3_20['20. If you park the car within the campus, have you any difficulty finding a parking space?'].notna()]
df3_20_train.shape

# null as testing dataset
df3_20_test = df3_20[df3_20['20. If you park the car within the campus, have you any difficulty finding a parking space?'].isnull()]
df3_20_test.shape
# print(df3_20_train.isnull().sum())

x_train_20 = df3_20_train.drop('20. If you park the car within the campus, have you any difficulty finding a parking space?',axis = 1)
x_train_20.shape

y_train_20 = df3_20_train['20. If you park the car within the campus, have you any difficulty finding a parking space?']
y_train_20
# print(x_train_20.isnull().sum())
# print('\n', y_train_20.isnull().sum())

from sklearn.ensemble import RandomForestClassifier
r_forest = RandomForestClassifier()
r_forest.fit(x_train_20, y_train_20)

# dataset excepts '18.need car for other purposes' feature with null values
x_test_20 = df3_20_test.drop('20. If you park the car within the campus, have you any difficulty finding a parking space?',axis = 1)
x_test_20

y_pred_20 = r_forest.predict(x_test_20)
y_pred_20

#creating a new dataset to store prediction values and ID position
df_null_20 = pd.DataFrame(y_pred_20, columns=['prevision'])

#reset index on null dataset
nan_20 = df3_20_test.copy().reset_index()
```

here we are predicting missing values of '21. What would you do if car parking was unavailable at the University?' using the Random Forest model. First, we separated the null value dataset as testing and the nonnull value dataset as the training dataset.

```
# dropping nan value columns except predicting column
df3_21 = df3.drop(df3.columns[[37,38,45]],axis = 1)
df3_21

# Splitting dataset into two i.e, non-null as training dataset and null as testing dataset
# non-null as training dataset
df3_21_train = df3_21[df3_21['21. What would you do if car parking was unavailable at the University?'].notna()]
df3_21_train.shape

# null as testing dataset
df3_21_test = df3_21[df3_21['21. What would you do if car parking was unavailable at the University?'].isnull()]
df3_21_test.shape
# print(df3_21_train.isnull().sum())

x_train_21 = df3_21_train.drop('21. What would you do if car parking was unavailable at the University?',axis = 1)
x_train_21

y_train_21 = df3_21_train['21. What would you do if car parking was unavailable at the University?']
y_train_21
# print(x_train_20.isnull().sum())
# print('\n', y_train_20.isnull().sum())

from sklearn.ensemble import RandomForestClassifier
r_forest = RandomForestClassifier()
r_forest.fit(x_train_21, y_train_21)

# dataset excepts '18.need car for other purposes' feature with null values
x_test_21 = df3_21_test.drop('21. What would you do if car parking was unavailable at the University?',axis = 1)
x_test_21

y_test_21 = df3_21_test[['21. What would you do if car parking was unavailable at the University?']]
y_pred_21 = r_forest.predict(x_test_21)
y_pred_21
```

```
#creating a new dataset to store prediction values and ID position
df_null_21 = pd.DataFrame(y_pred_21, columns=['prevision'])

#reset index on null dataset
nan_21 = df3_21_test.copy().reset_index()

#add column in the null dataset
df_prev_21 = pd.concat([nan_21, df_null_21], axis=1)

#set index
df_prev_21 = df_prev_21.set_index('index')

#fill the values
df3 = df3.fillna({'21. What would you do if car parking was unavailable at the University?':df_prev_21['prevision']}).copy()

# df3['21. What would you do if car parking was unavailable at the University?'] = lab_encoder.fit_transform(df3['21. What would you do if car parking was unavaila
df3.head(5)
```

here we are predicting missing values of '22. Are you aware of the Aston University car share scheme?' using the Random Forest model. First, we separated the null value dataset as testing and the nonnull value dataset as the training dataset.

]:
```
# dropping nan value columns except predicting column
df3_22 = df3.drop(df3.columns[[38,45]],axis = 1)
df3_22

# Splitting dataset into two i.e, non-null as training dataset and null as testing dataset
# non-null as training dataset
df3_22_train = df3_22[df3_22['22. Are you aware of the Aston University car share scheme?'].notna()]
df3_22_train.shape

# null as testing dataset
df3_22_test = df3_22[df3_22['22. Are you aware of the Aston University car share scheme?'].isnull()]
df3_22_test.shape
# print(df3_22_train.isnull().sum())

x_train_22 = df3_22_train.drop('22. Are you aware of the Aston University car share scheme?',axis = 1)
x_train_22
```

]:
```
# dropping nan value columns except predicting column
df3_23 = df3.drop(df3.columns[[45]],axis = 1)
df3_23

# Splitting dataset into two i.e, non-null as training dataset and null as testing dataset
# non-null as training dataset
df3_23_train = df3_23[df3_23['23. Which of the following would encourage you to car share either as a driver or passenger or improvements'].notna()]
df3_23_train.shape

# null as testing dataset
df3_23_test = df3_23[df3_23['23. Which of the following would encourage you to car share either as a driver or passenger or improvements'].isnull()]
df3_23_test.shape
# print(df3_23_train.isnull().sum())

x_train_23 = df3_23_train.drop('23. Which of the following would encourage you to car share either as a driver or passenger or improvements',axis = 1)
x_train_23

y_train_23 = df3_23_train['23. Which of the following would encourage you to car share either as a driver or passenger or improvements']
y_train_23
# print(x_train_23.isnull().sum())
# print('\n', y_train_23.isnull().sum())

from sklearn.ensemble import RandomForestClassifier
r_forest = RandomForestClassifier()
r_forest.fit(x_train_23, y_train_23)

# dataset excepts '18.need car for other purposes' feature with null values
x_test_23 = df3_23_test.drop('23. Which of the following would encourage you to car share either as a driver or passenger or improvements',axis = 1)
x_test_23

y_pred_23 = r_forest.predict(x_test_23)
y_pred_23

#creating a new dataset to store prediction values and ID position
df_null_23 = pd.DataFrame(y_pred_23, columns=['prevision'])

#reset index on null dataset
nan_23 = df3_23_test.copy().reset_index()
```

```
#fill the values
df3 = df3.fillna({'23. Which of the following would encourage you to car share either as a driver or passenger or improvements':df_prev_23['prevision']}).copy()

# df3['23. Which of the following would encourage you to car share either as a driver or passenger or improvements'] = lab_encoder.fit_transform(df3['23. Which of
df3.head(5)
```

here we are predicting missing values of '31. Which of the following would encourage you to travel here more by bicycle or improvements' using the Random Forest model. First, we separated the null value dataset as testing and the nonnull value dataset as the training dataset.

```
# Splitting dataset into two i.e, non-null as training dataset and null as testing dataset
# non-null as training dataset
df3_31_train = df3[df3['31. Which of the following would encourage you to travel here more by bicycle or improvements'].notna()]
df3_31_train.shape

# null as testing dataset
df3_31_test = df3[df3['31. Which of the following would encourage you to travel here more by bicycle or improvements'].isnull()]
df3_31_test.shape
# print(df3_23_train.isnull().sum())

x_train_31 = df3_31_train.drop('31. Which of the following would encourage you to travel here more by bicycle or improvements',axis = 1)
x_train_31

y_train_31 = df3_31_train['31. Which of the following would encourage you to travel here more by bicycle or improvements']
y_train_31
# print(x_train_31.isnull().sum())
# print('\n', y_train_31.isnull().sum())

from sklearn.ensemble import RandomForestClassifier
r_forest = RandomForestClassifier()
r_forest.fit(x_train_31, y_train_31)

# dataset excepts '18.need car for other purposes' feature with null values
x_test_31 = df3_31_test.drop('31. Which of the following would encourage you to travel here more by bicycle or improvements',axis = 1)
x_test_31

y_pred_31 = r_forest.predict(x_test_31)
y_pred_31
```

```
#reset index on null dataset
nan_31 = df3_31_test.copy().reset_index()

#add column in the null dataset
df_prev_31 = pd.concat([nan_31, df_null_31], axis=1)

#set index
df_prev_31 = df_prev_31.set_index('index')

#fill the values
df3 = df3.fillna({'31. Which of the following would encourage you to travel here more by bicycle or improvements':df_prev_31['prevision']}).copy()

# df3['31. Which of the following would encourage you to travel here more by bicycle or improvements'] = lab_encoder.fit_transform(df3['31. Which of the following

df3
```

The columns are reindexing here.

```
df3_f=df3.reindex(columns= ['2.member', '3.gender', '5.age','6.how far from home to workplace','9.mode of travel change based on season', '10. Why do you normally
```

creating another dataset from the before dataset.

```
df4 = df3_f.drop(df3_f.iloc[:, 18:29],axis = 1)
df4.head(5)
```

Feature selection

Here, Here, features are selecting as dependent and independent variables. we are predicting transport modes(those columns in Y) based on these features in X.

```
X = df4.iloc[:, 0:8]
Y = df3_f[['7.5.f. Bus - Everyday','7.2.f. Rail - Everyday','7.3.f. Metro - Everyday','7.4.f. Walk - Everyday','7.5.f. Cycle - Everyday','7.6.f. Drive all the way
```

Split the dataset for training and testing.

Split the dataset for training and testing.

```
xtrain, xtest, ytrain, ytest=train_test_split(X, Y, train_size=0.7, random_state=0)
```

First, we have to build machine learning model with training data. Here we are using multioutput classifier ensemble with RandomForest classifier.

```
clf = multioutput.MultiOutputClassifier(ensemble.RandomForestClassifier())
clf.fit(xtrain,ytrain)
```

```
y_predi = clf.predict(xtest)
y_predi
```

Calculating accuracy to evaluate that Random Forest model is working well or not.

```
accuracy_models = np.sum(np.equal(ytest, y_predi)) / len(ytest)*100
accuracy_models
```

creating another dataset from the before dataset.

```
df5 = df3_f.drop(df3_f.iloc[:, 8:18],axis = 1)
df5.head(5)
```

Feature selection

Here, features are selecting as dependent and independent variables. we are predicting transport modes(those columns in Y) based on these features in X.

```
X_x = df5.iloc[:, 0:8]
Y_y = df5[['8.1.f. Bus - Everyday','8.2.f. Rail - Everyday','8.3.f. Metro - Everyday','8.4.f. Walk - Everyday','8.5.f. Cycle - Everyday','8.6.f. Drive all the way
```

Split the dataset for training and testing.

Split the dataset for training and testing.

```
xtrain_t, xtest_t, ytrain_t, ytest_t=train_test_split(X_x, Y_y, train_size=0.7, random_state=0)
```

First, we have to build machine learning model with training data. Here we are using multioutput classifier ensemble with Gradient boosting classifier.

```
clf_t = multioutput.MultiOutputClassifier(ensemble.GradientBoostingClassifier())
clf_t.fit(xtrain_t,ytrain_t)
```

```
y_predi_t = clf_t.predict(xtest_t)
y_predi_t
```

Calculating accuracy to evaluate that Gradient boosting model is working well or not.

```
accuracy_models_t = np.sum(np.equal(ytest_t, y_predi_t)) / len(ytest)*100
accuracy_models_t
```

```

```