Task1: Exploratory Data Analysis (EDA)

We have a dataset with the indices of Debit card transaction data. This dataset contains 1447 rows and 9 columns. The meaning of their attributes and data type are shown in Table 1.

Table 1. Summary of attributes of dataset

#	Column name	Meaning	Data type
0	Date	Date of transaction	Datetime: year/month/day
1	80% of the expenses were made by	80% of the expenses were made either by private/public sector	Object, Nominal
2	80% of the expenses were made by .1	80% of the expenses were made by Male/Female/Other	Object, Nominal
3	80% of the expenses were made in the following regions	The regions in which the 80% of the expenses were made	Object, Nominal
4	Miscellaneous (£)	The money spent on miscellaneous expenses	Float, numerical continuous
5	Automotive Fuel (£)	The money spent on automotive fuels	Float, numerical continuous
6	Entertainment (£)	The money spent on entertainment	Float, numerical continuous
7	Food and Drink (£)	The money spent on food and drink	Float, numerical continuous
8	Pubs, restaurants and fast food (£)	The money spent on pubs, restaurants, and fast food	Float, numerical continuous

One of the first things I thought to check was to see if the data had some meaning in time. I noticed that it was not in chronological order, but after sorting, it was complete in time. The information ranges from the 1st of January 2020 to the 17th of December 2023.
Then, I checked if there were any missing values in the columns. As seen in Figure 1, just 4 attributes have null values, and there are no rows with a significant large number of missing values, apparently just 4 have 2 missing values. Of those 4 columns, 2 of them are categorical and 2 are numerical.

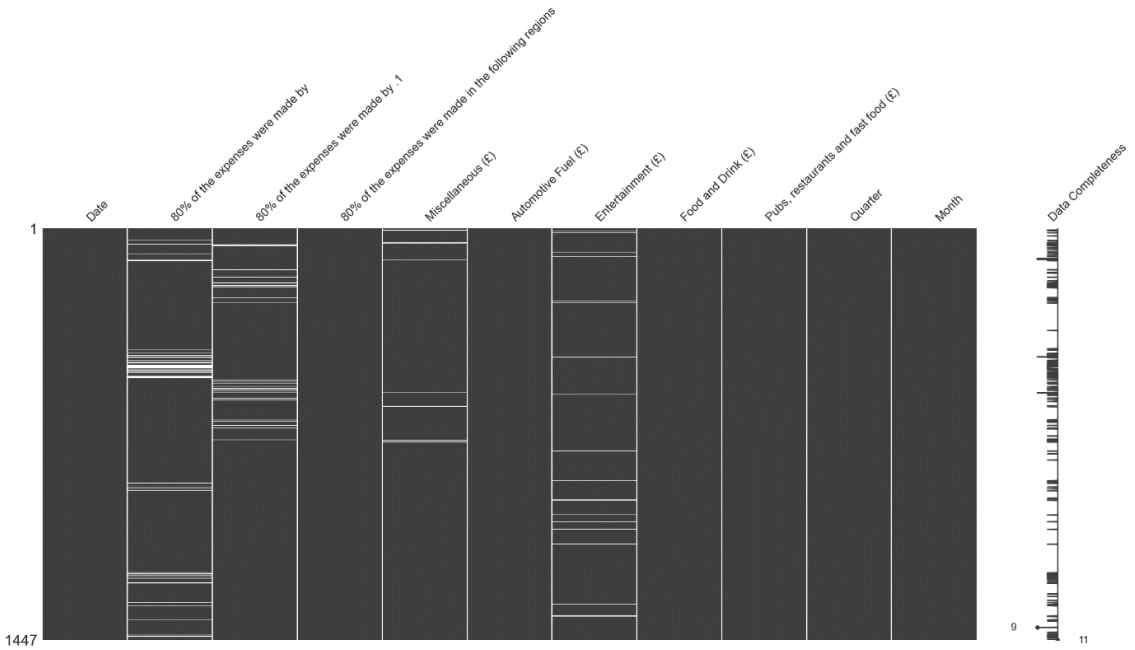

Figure 1. Visualising missing data in the dataset

Figure 2 shows the histograms of the categorical features, which is useful to see which values are contained in those variables.

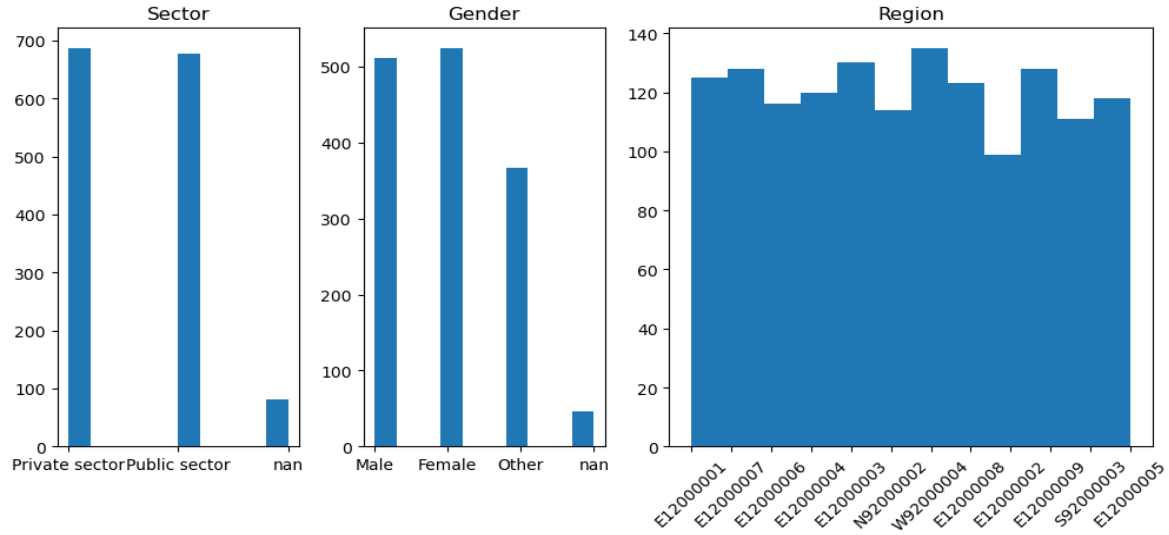

Figure 2. Histograms of categorical features

Figure 3 shows the distributions of the numerical features with boxplots. There is an outlier in 'Automotive Fuel (£)' attribute. Miscellaneous looks kind of normal, as automotive fuel. The other 3 look skewed.

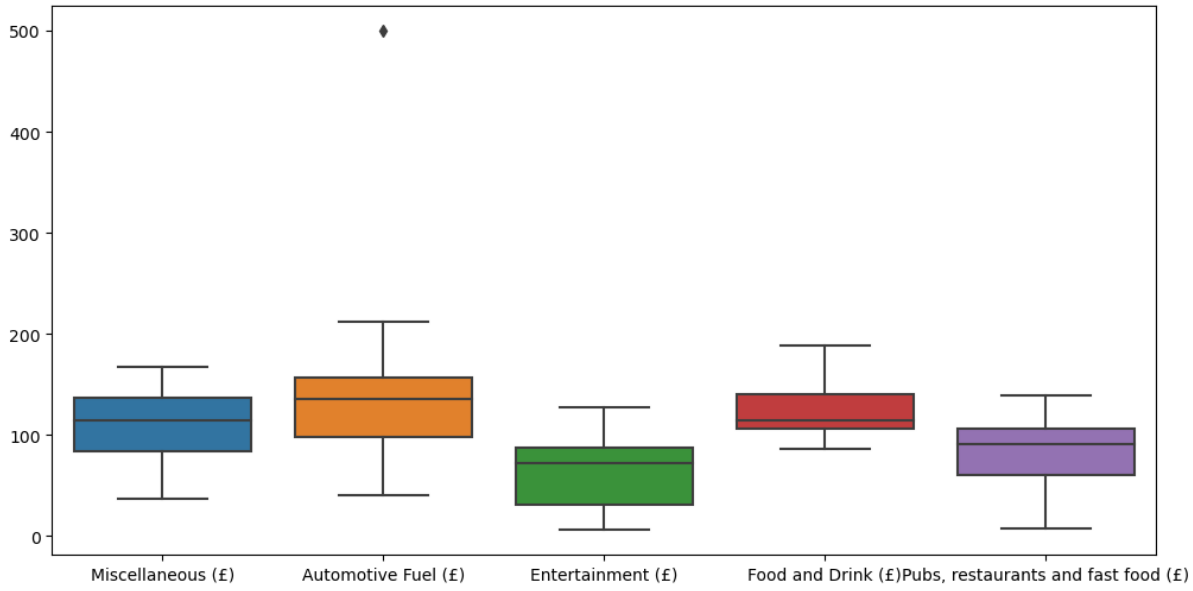

Figure 3. Box plots of numerical features

So far, we know that from the numerical features we need to handle missing values to 2 columns and an outlier in one column. I plotted these columns in Figure 4, to have a better idea of the data.

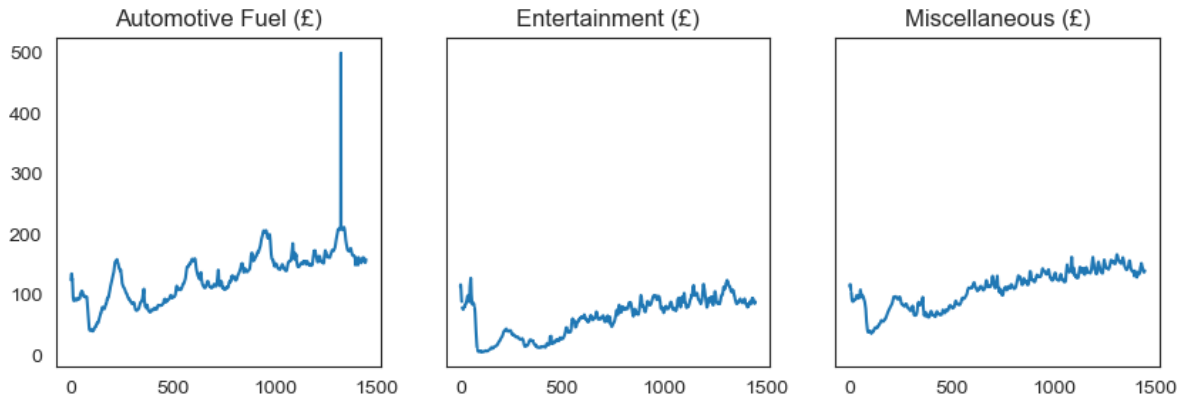

Figure 4. Line plots of columns that need handling.

Finally, I plotted the median by month of the numerical variables to see tendencies. All variables have increased over time. “Automotive fuel”, “Pubs, restaurants and fast food” and “Entertainment” have sharp peaks, during the summer months. “Miscellaneous” also follows those peaks in less measure. ‘Food and Drink’ varies less than the others, so it must refer to the expenses in groceries and household staples, which are less prone to seasonality.

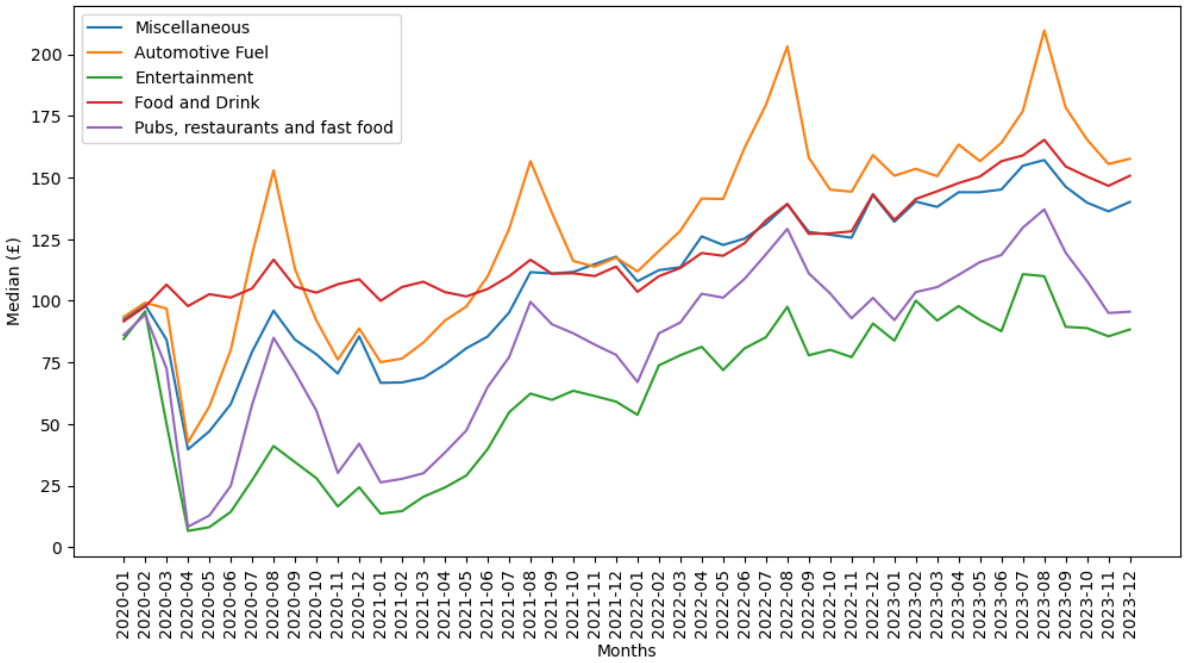

Figure 5. Median of expenses per month, of numerical features

Task 2: Data Pre-processing

The first thing to do was to change the name of the attributes because they were confusing or too long (Table 2).

Table 2. Renaming columns

Column name	Change to
80% of the expenses were made by	Sector
80% of the expenses were made by .1	Gender
80% of the expenses were made in the following regions	Region
Miscellaneous (£)	Miscellaneous
Automotive Fuel (£)	AutomotiveFuel
Entertainment (£)	Entertainment
Food and Drink (£)	Groceries
Pubs, restaurants and fast food (£)	EatingOut

Second, I decided to eliminate the row containing the outlier in "Automotive Fuel" because it was distorting the data and wasn't reliable.

For the handling of missing values, I did the following:

- Erased the rows with 2 missing values, because those rows were not reliable.
- Replaced the missing values with the word "Missing" in the categorical features. Being 3% and 5% of the data, I didn't want to erase them (Table 3).
- Filled the numerical features with an interpolated value. This way the data won't vary much from its original distribution.

Table 3. Handling missing values

Column with missing values	% of missing values	Handling
Sector	5%	Replace with word "Missing"
Gender	3%	Replace with word "Missing"
Entertainment	2%	Replace with interpolation
Miscellaneous	1%	Replace with interpolation

Figure 6 shows how the data clearer than Figure 4 and is understandable as the data is sorted in time.

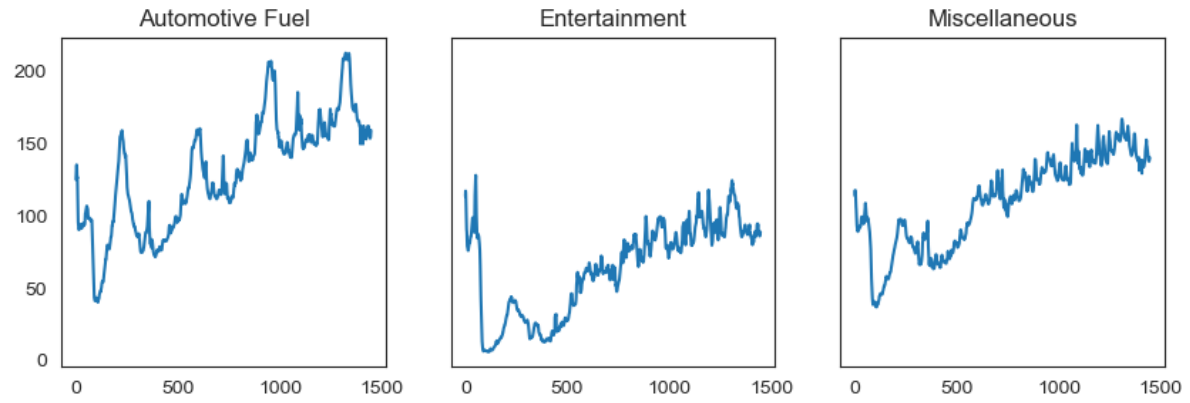

Figure 6. Line plots of numerical features after handling

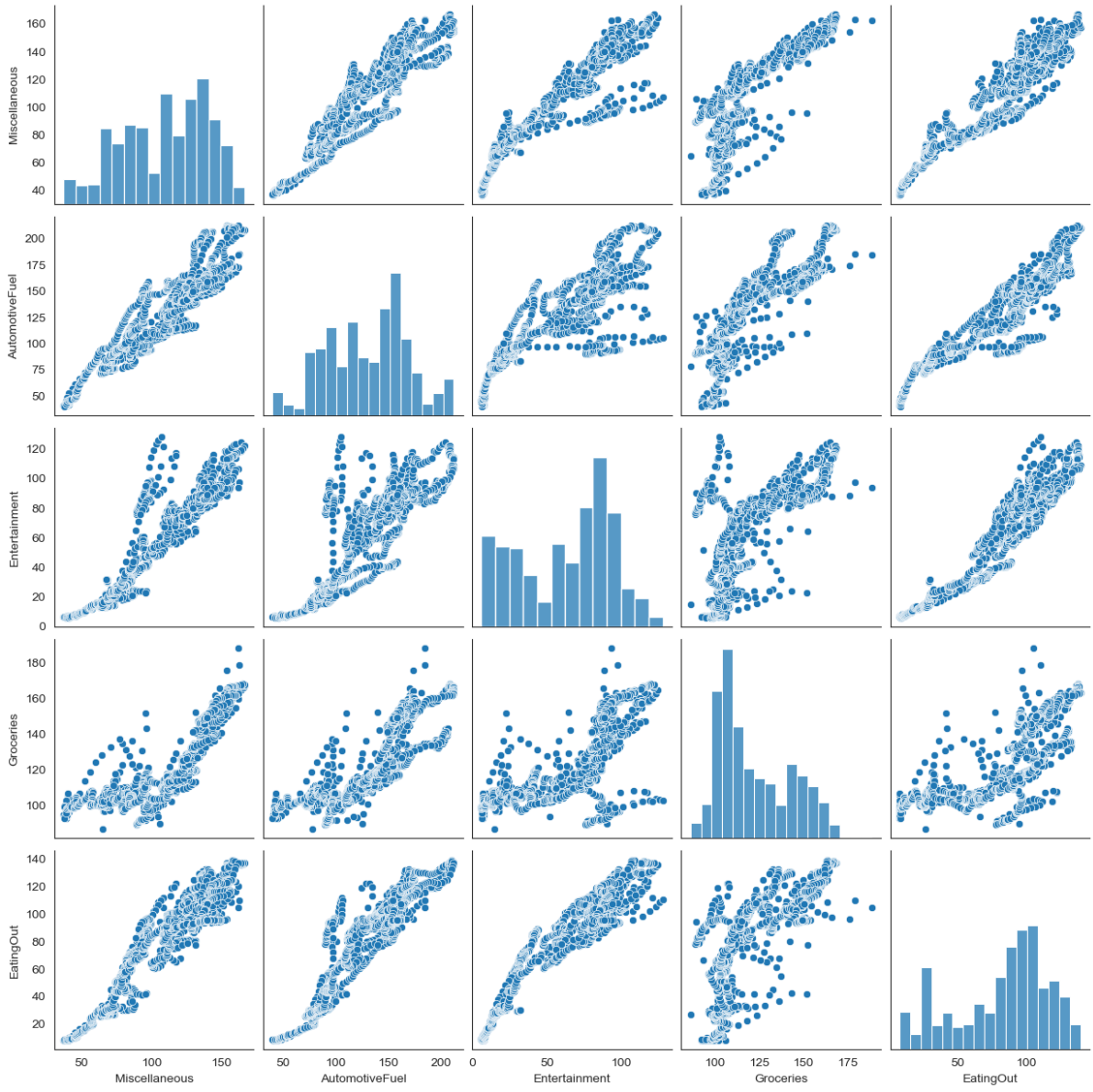

Figure 7. Pair plot of numerical features after data cleaning

In Figure 7 is easier to see that some variables are correlated: Like Miscellaneous, Entertainment and Automotive Fuel. Groceries is the least correlated.

The encoding of categorical features I explain in Table 4.

Table 4. Encoding

Column	Encoding technique	Reason
Sector	One-hot encoding	It only has 3 values (counting "Missing") and there is no hierarchy in them.
Gender	Label encoding	I could have done one-hot encoding, but I needed to use it later as a binary column, so I decided with label encoding would be easier to erase the rows I wouldn't need.
Region	Label encoding	It has 12 values, so one-hot encoding would produce a very sparse data frame.

Finally, I did the normalisation of numerical features because I needed to work on different models, and I thought that having all the features normalised would make it easier to compare and less prone to bias. I used the z-score normalisation to preserve the shape of the distributions. The final dataset can be seen in Figure 8.

```
# New dataframe
db6 = pd.concat([db_num, enc_features], axis=1)
```

```
db6.head(10)
```

3]:

	Miscellaneous	AutomotiveFuel	Entertainment	Groceries	EatingOut	GenderCode	MissingS	PrivateSector	PublicSector	RegionCode
0	0.134839	-0.134307	1.604683	-0.998047	1.082282	0	0.0	1.0	0.0	0
1	0.251187	-0.042499	1.718644	-0.727632	1.167512	1	0.0	1.0	0.0	6
2	0.255054	0.052690	1.674272	-0.716736	1.168399	1	0.0	1.0	0.0	5
3	0.211222	0.130453	1.444117	-0.695439	1.095599	1	0.0	1.0	0.0	3
4	0.130005	0.126032	1.153310	-0.739518	0.910935	1	0.0	1.0	0.0	2
5	-0.049190	0.009257	0.933050	-1.072832	0.658206	1	0.0	1.0	0.0	10
6	-0.116710	-0.110119	0.865376	-1.601777	0.358718	1	0.0	1.0	0.0	11
7	-0.184230	-0.089053	0.822920	-1.325914	0.155410	0	0.0	0.0	1.0	3
8	-0.334096	-0.253942	0.664748	-1.449235	0.012769	1	0.0	1.0	0.0	3
9	-0.457534	-0.498416	0.506576	-1.510648	-0.068614	2	0.0	0.0	1.0	11

```
db6.shape
```

9]: (1441, 10)

Figure 8. Dataset after pre-processing.

Task 3: Supervised ML algorithm: Prediction

We need to predict the outcome of one attribute ("Miscellaneous") from other 7 attributes, 3 of them encoded in 5 columns. For this task, I started with a multiple linear regression because it was the easiest to understand and apply. But because the linear regression will consider all the variables have the same importance, I decided to try the Lasso regression (Least absolute shrinkage and selection operator regression), which adds a bias to the equation, giving more importance to the variables that affect more the target. Finally, out of a personal interest, I wanted to try an Artificial Neural Network model, because it could be the case that the variables don't have a linear relationship.

For the multiple linear regression and the Lasso regression I used the sklearn linear model module, which we used in Tutorial 4.
For the Artificial Neural Network, I used TensorFlow library, following a tutorial online (Srivignesh, 2023). Here we used 4 layers (160, 480, 256 units and the output) with ReLU activation function for the model. For the loss function we used the Mean Squared Logarithmic Loss and Adam as optimizer. For the training we used 10 epochs and a batch size of 64. With this model we check the history of the mean squared logarithmic error, reducing with each epoch, so I saw that the model was working on my data. It also changed when changing the batches and epochs, but I could not spend more time trying to find a better fit.

Table 5 shows part of the results after running the models, with the same test section of the data. There are little differences between them.

Table 5. Predicted and actual values of the Miscellaneous expenses, for each model.

Multiple Linear Regression		**Lasso Regression**		**Artificial Neural Network**	
Y predicted	Y test	Y predicted	Y test	Y predicted	Y test
[108.69 99.32]		[108.9 99.32]		[107.9 99.32]	
[134.96 150.01]		[135.1 150.01]		[134.96 150.01]	
[146.31 144.56]		[145.67 144.56]		[145.07 144.56]	
[71.32 86.72]		[71.46 86.72]		[69.2 86.72]	
[58.61 51.37]		[58.49 51.37]		[54.32 51.37]	
[65.19 67.73]		[65.45 67.73]		[64.28 67.73]	
[97.57 94.89]		[97.3 94.89]		[97.24 94.89]	
[81.43 82.58]		[81.33 82.58]		[80.33 82.58]	
[72.27 72.88]		[72.2 72.88]		[71.37 72.88]	
[136.37 144.5]		[136.48 144.5]		[136.09 144.5]	
[102.33 114.6]		[102.23 114.6]		[101.92 114.6]	
[103.59 111.94]		[103.8 111.94]		[103.43 111.94]	
[65.26 58.2]		[64.57 58.2]		[62.29 58.2]	
[100.68 114.61]		[100.91 114.61]		[100.39 114.61]	
[140.92 138.37]		[140.73 138.37]		[140.54 138.37]	
[83.53 82.]		[83.18 82.]		[82.71 82.]	
[108.79 121.96]		[109.03 121.96]		[109.3 121.96]	
[127.09 126.67]		[127.34 126.67]		[129. 126.67]	
[122.47 124.85]	…	[122.65 124.85]	…	[123.09 124.85]	…

To see how far the predicted values are from the real test values in each model, I run the Mean squared error (MSE), the Root mean squared error (RMSE), the Mean absolute error (MAE) and the R squared (R^2). Table 6 shows the results of these. These evaluation metrics, apart from the R squared, have similar equations. But the MSE being squared shows a larger number, which is useful with small

variations. These measurements of error indicate that the lower one is the most accurate, in this case the Lasso regression.
The R squared measures the percentage of variance in the dependent variable, which is explained by the independent variables, being a percentage is easier to read.
Again, in this case the Lasso regression shows a larger percentage (94.76%), comparing with the other models (94.71% and 94.28%) so it is slightly a more reliable model, although the differences are small.

Table 6. Evaluation metrics

Model	MSE	RMSE	MAE	R^2
Multiple linear regression	48.1539	6.9393	5.3100	0.9471
Lasso regression	47.6458	6.9025	5.2299	0.9476
Artificial neural network	52.0718	7.2160	5.5883	0.9428

Figure 10 shows the graphs of the predicted values with the test values for each model. All of them show a good fit for our data.

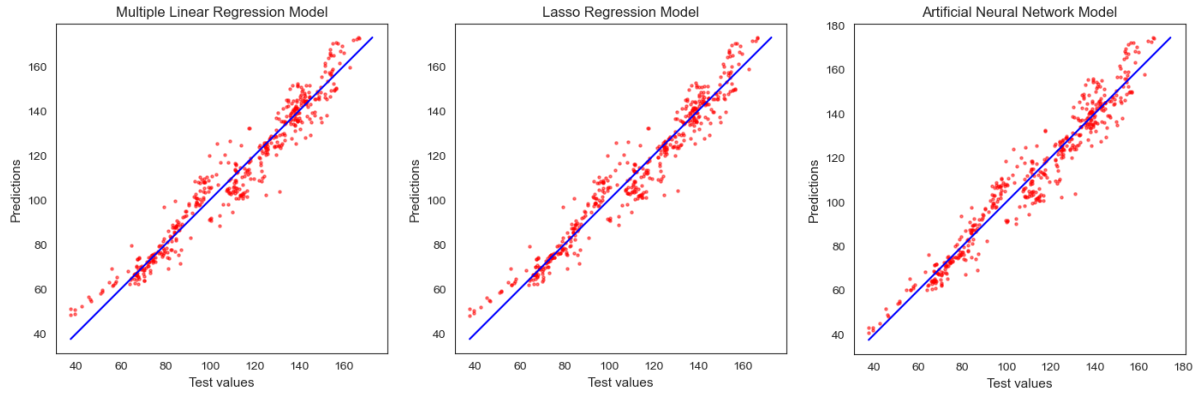

Figure 9. Scatter plots of the predicted and test values of the 3 models.

This means that our data could be predicted with the Multiple linear regression without many complications, because the computing resources of using the other models, especially the artificial neural networks, are not justified.
As an additional observation, when I first tried the models, I run them with the normalised data. This showed the same result for the multiple linear regression because it just works in proportion. But it was notoriously less accurate in the artificial neural network regression. I believe this is because the hyperparameters needed to be adjusted so the little variations are better perceived. Furthermore, I would have like to try the models changing the encoded variables, to see if there is any difference between label and one-hot encoding.

Task 4: Supervised ML algorithm: Classification

For this task we need to predict the gender given the 5 columns of expenses (Miscellaneous, Automotive Fuel, Entertainment, Groceries, and Pubs, restaurants, and fast food). After filtering the rows with missing or other values, we have a binary classification (Male = 0, Female = 1). Following the examples in class, I performed the Logistic Regression, Naïve Bayes classification and Decision trees, to see how they compare, with the normalised dataset.
The logistic regression is the easiest to perform and understand, based on our independent variables the algorithm elaborates a mathematical formula to set a threshold to estimate how probable the target would be on each side of that threshold.
The Naïve Bayes classification is easy to implement in python with sklearn naive_bayes. It saves us the work of studying the dataset (train) to find the combination of probabilities of all the variables in which the gender is male or female. It will assume that all our variables are independent, or not correlated.
Decision trees classification in machine learning will compute a combination of binary decisions following all the variables in our training dataset, to learn to classify the target. The formulas behind these decisions are not very clear, but we can get the hierarchy of the features its considering. In this case, it found that "Automotive fuel" was the feature with more importance to predict gender although closely followed by the other features (Figure 10).

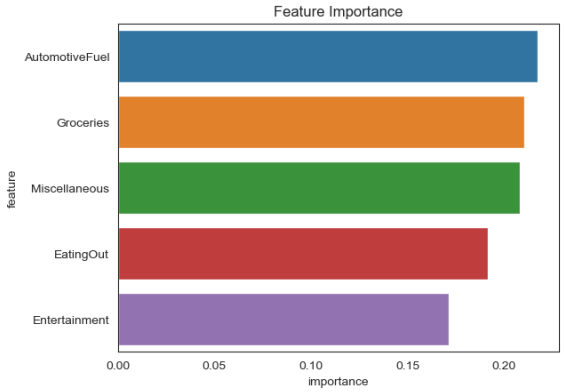

Figure 10. Feature importance in decision tree algorithm

The accuracy scored, that is the sum of true positives and true negatives divided by the sum of all results, is in Table 7, showing that the decision trees model (0.58) was accurate more times than the logistic regression model (0.50) and the Naïve Bayes model (0.45). Overall, these are not great values, if we round them up, all the models are correct between 45% and 58% of the time. But these definitely would be determinant to choose a model.

Table 7. Accuracy scores of classification models

Logistic regression	Naïve Bayes	Decision trees
0.50775	0.45348	0.58139

I chose to compare the confusion matrices because is easy to read and understand.
Looking closer to the confusion matrices (Figure 11), we find:

- Logistic regression: of 126 males in the test (0 values) only guessed 55 (43%), and of 132 females (1 values) it found 76 (57%)

- Naïve Bayes: of 126 males in the test (0 values) only guessed 30 (23.8%), and of 132 females (1 values) it found 87 (65.9%)
- Decision trees: of 128 males in the test (0 values) it guessed 74 (57.8%), and of 130 females (1 values) it found 76 (58.4%)

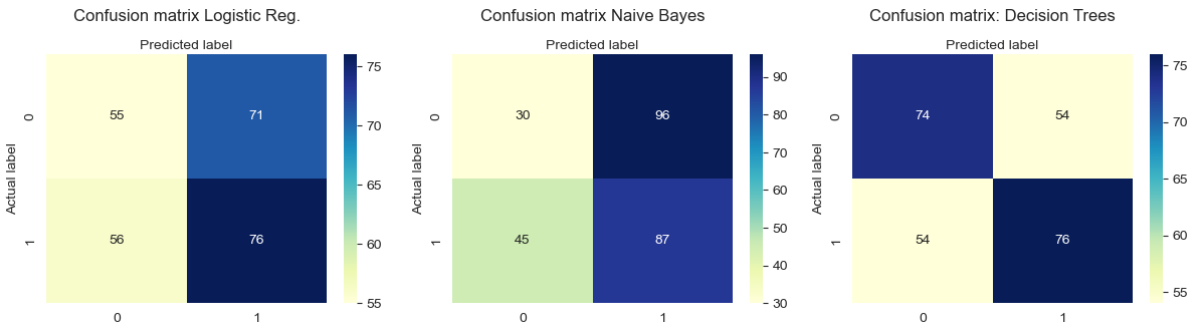

Figure 11. Confusion matrix of each model for classification

The Naives-bayes classifier assumes that all variables are independent. Being the worst performer of all the classifiers may be due to the non-independence of the variable used.
In general, the fact that all classifiers showed poor performance may indicate that the variables used are not adequate to predict the gender.

Task 5: Unsupervised ML algorithm

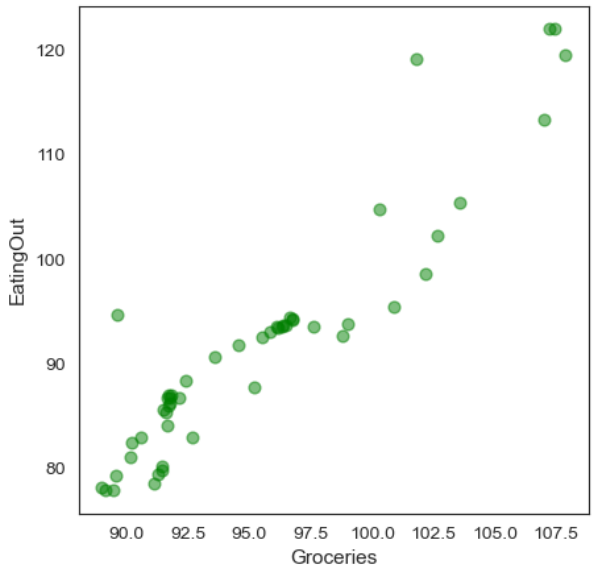

Figure 12. Scatter plot of case study

For this task I chose the features "Groceries" and "EatingOut" because I thought it would be interesting to find if there are similar expenditure behaviours. Figure 12 shows the first 50 tuples of those features in a scatter plot.

- **What unsupervised algorithm would you prefer to apply and mention why have you chosen that?**
  I chose the K-means algorithm because it is easy to understand how it works. It calculates the distance between each data point and a centroid to assign it to a cluster, then the mean of those distances is the new centroid. It repeats the process with the new centroid, until it doesn't change.

- **What could be other alternative algorithms that could be applied?**
  Another algorithm is the Density-based spatial clustering of applications with noise (DBSCAN). This algorithm is good to cluster points that present high density and can be separated of the other clusters by areas of low density. It can identify clusters of any shape and outliers, these are noise, or points in the low-density areas (Géron, 2023).

- **Into how many clusters you would partition them?**
  The data looks like correlated, but for interest, I would separate it in 3 clusters. One group of those with low spent in groceries and in eating out, a second group with medium spent in groceries and eating out, and the third group with high spent in groceries and eating out.

- **How do you arrive at the optimal number of clusters?**
  I use the elbow method to see how the within-cluster-sum-of-square values changes when increasing the number of clusters. In case of not clear inflexion point, I would have to check the silhouette score.
  In our case of study, in Figure 13, we see the point of inflexion in k = 3 in the elbow graph, but when I find the silhouette score, I found that k = 2 is the highest, so it could be the best option. Our data does look like we could just separate it in low expenders and high expenders.

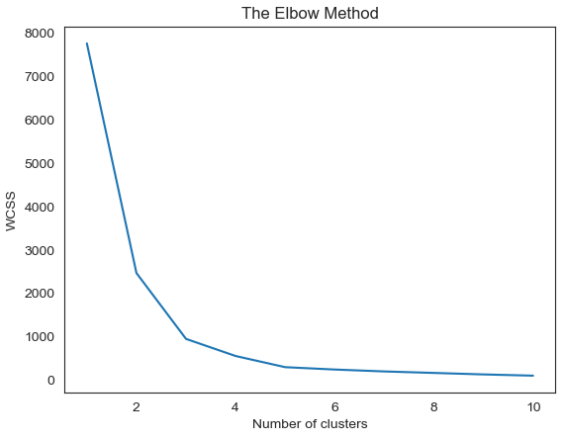

Figure 13. Elbow method graph.

Table 8. Silhouette score.

K	Silhouette score
2	0.67317
3	0.61623
4	0.61627
5	0.62116
6	0.57786

- **Would you prefer to do a trial-and-error method or rather apply any relevant algorithms for arriving at the optimal number of clusters.**
  Depending on the data, sometimes it would be better to find an algorithm for the optimal number of clusters, but in some cases, a trial-and-error method could be part of the data exploration.
  In our case study, a trial-and-error approach was useful to see how the algorithm is working. I found that for this dataset, K-means is not the most reliable method for clustering.
  Figure 14 shows the results of trying different numbers of clusters in our data.

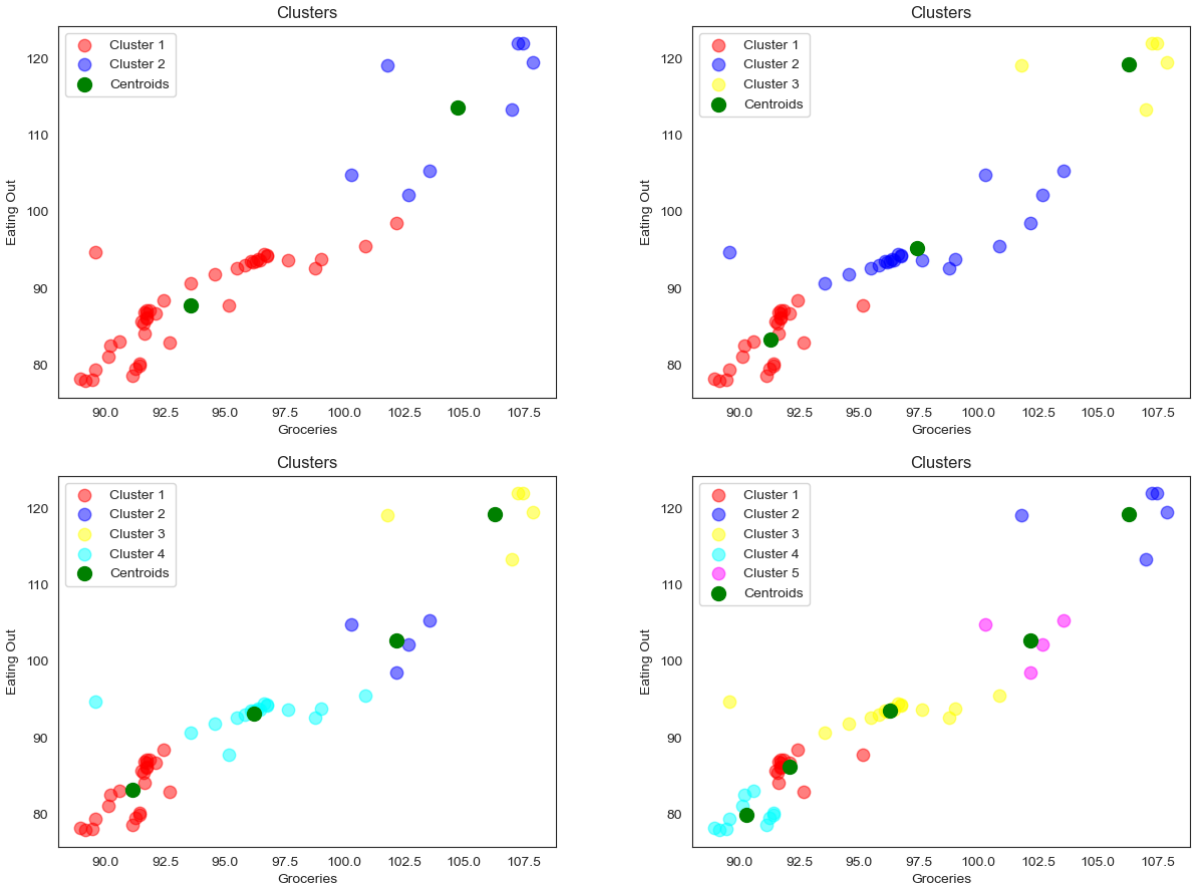

Figure 14. Trying different k numbers to cluster the points.

Task 6 – Conclusion

In this coursework we used a structured dataset of 1447 records. I tried to follow all the class tutorials to do all the tasks, including what we previously learned in the programming module. In some cases, I had to check online for other examples of how that task was done, to understand properly so I could make the code work.
Tasks 1 and 2 were worked cyclically, that is while working the following tasks, I found I needed some or other information I wasn't showing. For example, I didn't think a correlation analysis would be needed to understand the data, because the categories didn't seem dependent on each other to me. But then to think about the linear regression, this was something to look at. However, to do the correlation analysis I noticed that filling missing numerical values, the media or mean didn't work. I had to interpolate them, to make sure the imputed data wasn't disturbing the features correlations.
For task 3: supervised algorithms for regression, I did a bit of research to find the difference between some regression algorithms. I used the Multiple linear regression as learned in class, the Lasso regression, which I found was always mentioned as an improved version of multiple linear regression and was quite easy to implement, and the Artificial Neural Network algorithm. This last one was discussed in class but was used for classification, being a popular algorithm, I wanted to see how to make it work for prediction. However, it wasn't much efficient than the other algorithms in our case study, it may be worthy for bigger datasets.
For task 4: supervised algorithms for classification, I did the 3 algorithms taught in class: Logistic, Naïve Bayes and Decision trees. These 3 algorithms work quite differently mathematically and showed that for our dataset the Decision Trees works best. It would be interesting to see the result with more data, because when filtering the missing or other gender records, we get around 1000 records to work with.
For task 5: unsupervised algorithm, I just followed the example from the tutorial, but, after reading a bit more about it, I think the DBSCAN algorithm would have worked better with our data.
Overall, the coursework has been a very interesting work, to understand the application of different algorithms of machine learning.

References

Géron, A. (2023). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.* California: O'Reilly Media.

Srivignesh, R. (2023, July 18). *A Walk-through of Regression Analysis Using Artificial Neural Networks in Tensorflow*. Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/08/a-walk-through-of-regression-analysis-using-artificial-neural-networks-in-tensorflow/