**Introduction:**

This report aims to cover the process of data wrangling, by utilizing different data wrangling and exploration techniques, to conduct data exploration, preparation and transformation on a real-life data set.

Data Wrangling consists of Exploratory Data Analysis (EDA), Quantitative Analysis and Feature Engineering. EDA and Quantitative Analysis involves investigating and understanding the data, to see which variables are useful for making predictions. On the other hand, Feature Engineering involves cleaning, structuring and improving raw data/features into an appropriate format. These steps help to prepare the data for predictive modelling, evaluation through a linear regression model, and for machine learning algorithms. The effectiveness of these models are very dependent on the features that go into the model, as well as the way they are represented. Some representations can improve model effectiveness while others may be irrelevant and can compromise the model’s effectiveness.

For the scenario in this assignment, data from numerous supermarkets have been collated into a CSV (Comma-separated values) file. The management teams of these supermarkets seek to understand the different properties of their products and outlets, from the data through the data wrangling process, to further increase their sales.

The report will explore the various data transformation techniques used on the dataset, the model results/performance obtained from each of the different methods, and why certain methods were chosen for the final model result over others. The purpose of experimenting with different transformation methods is to compare each method's performance/results to determine which are the most appropriate and effective based on the model results.

**Data Exploration (EDA):**

Before we perform data and feature transformation, we first have to perform Exploratory Data Analysis (EDA) on the dataset, to help us investigate and observe the various relationships or discover patterns between the different variables/features, using a combination of univariate, bivariate and multivariate analysis to do so. This is done to improve our overall understanding of the dataset, and to identify useful features for making predictions.

I have created some exploratory questions to assist me in visualizing the data. These questions help act as tools or general guidelines to focus on a specific portion of the dataset and decide which visualization to use. These exploratory questions are,

* Which Item Type generated the most sales?
* Do Low Fat items generate more sales compared to regular items?
* Does Item Weight and Item Type have an impact on sales?
* Does Outlet Size affect sales?
* Is there any correlation between Item Sales and Item Visibility?
* Is there any correlation between Item Sales and Item MRP?
* Is there any correlation between Item Sales and Item Weight?
* Which item type has the highest visibility among the outlet types?

**Which Item Type generated the most sales?**

The first exploratory question is “Which Item Type generated the most sales?”. This is to compare the sales of the various item types in the supermarkets. For this question, a tree map visualization was used. A tree map visualization is used to show data as part of a whole, in this case, the sales for each item type is represented by a block in the tree map. The colour of the block represents the item type, while the size of each block denotes the sum of sales for that item type. This makes it easy to observe and compare the sales of each item type as part of the total sales.

Based on this tree map visualization, Fruits and Vegetables have the highest share of the total sales among all the item types.

**Do Low Fat items generate more sales compared to regular items?**

The second exploratory question is “Do Low Fat items generate more sales compared to regular items?”. The purpose of this question is to compare the sales of the items based on the fat content of the items. The visualization used here is a vertical bar chart. A bar chart is used as we want to perform a comparison for numerical values across a categorical variable. By using a bar chart, we are able to easily see which fat content group has the highest sales, and simultaneously compare them to other groups in the chart based on the size of each bar and its colour.

Based on the bar chart for this exploratory question, Low Fat items have the highest total sales among the other items.

**Does Item Weight and Item Type have an impact on sales?**

The next exploratory question is “Does Item Weight and Item Type have an impact on sales?”. This question aims to find out if item weight, based on a specific item type, has an impact on sales. Since we are analysing two variables simultaneously, we can use a dual axis chart to visualize these variables. The reason being, we can use two axes, represented by a line or a bar, to easily show the relationship between two different numerical variables with different magnitudes grouped by a categorical variable. This allows us to observe the relationship and identify any trends between the variables.

Based on the dual axis chart above, we can say that the variables do not have much impact on each other. For example, when item weight is high for an item type, it does not necessarily mean the sales for that item type will be high as well.

**Does Outlet Size affect sales?**

The next exploratory question, “Does Outlet Size affect sales?”, wants to find out if the size of the supermarket outlets has any impact and its sales. A pie chart visualization is used here as, similar to the tree map, we are interested in comparing different groups of the sales data as part of a whole. The size and colour of each segment represents the magnitude of sales and the outlet size respectively.

Based on the pie chart above, Medium outlets have the highest sales.

**Scatter Plots**

Next, I will cover the scatter plots used in the exploratory analysis and the question each plot relates to. A scatter plot is used to determine whether or not two numerical variables have a correlation with each other. For this assignment, the target variable is “Item\_Outlet\_Sales”. Therefore, we are interested in finding out how the other numerical variables affect the target variable, and if there is a positive or negative correlation between them. We can determine the correlation by observing the plots and using a line of best fit.

The first scatter plot answers the question, “Is there any correlation between Item Sales and Item Visibility?”. Based on the scatter plot and the line drawn, there is a weak negative correlation.

The next scatter plot answers the question, “Is there any correlation between Item Sales and Item MRP?”. From the scatter plot and the line drawn, there a weak positive correlation.

The final scatter plot answers the question, “Is there any correlation between Item Sales and Item Weight?”. Based on the scatter plot, there is no correlation between the variables.

**Which Item Type has the highest visibility among the outlet types?**

For the next exploratory question, “Which Item Type has the highest visibility among the outlet types?”, a stacked bar chart is used. A stacked bar chart is similar to a regular bar chart, allowing us to make comparisons between variables. However, a stacked bar chart also shows a larger category divided into smaller groups as part of the total amount, allowing us to compare each part of the bar.

In the bar chart above, outlet type is denoted by colour, and we can see that Fruits and Vegetables has the highest visibility in Supermarket Type 1.

**Cleaning the Data (before split & after)**

Data cleaning the an important process of data wrangling. In data wrangling, data cleaning involves removing any duplicates, removing outliers, fixing structural errors and filling in any missing data. This is important to ensure the machine learning models are not affected negatively by missing data or outliers, as leaving them in can cause several issues with the models, and return inaccurate results. I will be covering how I handled the missing values and outliers in the dataset.

**Outliers**

Handling outliers is an important part of data pre-processing for data wrangling. It is important to handle outliers found in the dataset, as they increase the variability in the dataset which also increases the error variance. This can negatively affect the model’s results and reduce our model’s accuracy.

Before handling the outliers, I have to first examine the distribution of the numerical variables to look for the ones that contain outliers. This was done by using a user defined function to create a diagnostic plot which consisted of three visualizations, a histogram, a Q-Q plot and a box plot. From the diagnostic plots, only “Item\_Outlet\_Sales” and “Item\_Visibility” are shown to contain outliers. Therefore, outlier handling will only be performed on these two variables.

For outlier handling, I used capping/censoring to cap the variable at an arbitrary maximum and minimum values. This was done using feature engine using the Winsorizer function. Then, defining the “gaussian” capping method for a normal distribution as well as specifying capping for both tails. This replaces the extreme values with values that are closer to the other values in the variable. Other capping methods such as “quantiles” and “iqr” did not improve model performance, and thus were not used.

Outlier Trimming was not used as I felt that removing the outliers outright would cause me to lose a large amount of data that would later affect the model results.

Winsorization and Zero Coding were tested with the dataset, but both of these methods ended up returning a worse model performance based on the Root Mean Square Error (RMSE) and R-Square results. Therefore capping/censoring was chosen as the preferred method to handle the outliers in this instance.

**Missing Values**

The missing values in the dataset have to be fixed and replaced before we can perform numerical and categorical transformations. I will be replacing the missing data using imputation methods instead of removing them, as replacing them with a substitute value allows for more of the data and information of the dataset to be retained, which can be important for accurate test results when running the data into a model later on.

Firstly I will cover the missing values that were handled before performing the train test split. Before performing the train test split on the dataset, I had to replace missing values for the variable “Item\_Weight”. This was done to be able to perform numerical transformation on the dataset before the train test split, as transformation is not possible when there are null values in any of the variables. To replace the values for this variable, I used .fillna() to fill the null values in the variable with an arbitrary number.

Next, I will cover how the missing values were handled after performing the train test split. At this point, when checking the features, the only features that contained missing values was “Outlet\_Size”. To replace the missing values in the feature, I used a feature engine pipeline to impute missing categorical values into the feature. The method used for the imputation pipeline was to replace the null values with the word “Missing”. This was done by defining a list of the features with missing values, them imputing the list into the pipeline and use the CategoricalImputer to impute the missing values. Replacing the missing values with a frequent value did not show any improvements the model results at all, so I opted for the missing value indicator method used previously.

**Data Transformation (categorical and numerical)**

Data transformation is the process of converting data stored as one format or structure to another format. This is to ensure that data would be suitable for modelling later on. I performed data transformation on the categorical as well as the numerical variables.

**Categorical Encoding**

Categorical encoding consists of converting categorical data into an integer format so that the models will be able to understand and utilize the data, as machines are unable to understand categorical data.

To perform categorical encoding, I performed one-hot encoding of frequent categories to encode the categorical variables as binary variables, while also reducing the number of these binary variables. This was done with feature engine, using the OneHotEncoder function and indicating top categories = 5, which means that the encoding will only be done on the most frequent 5 categories. This is to reduce the number of binary variables that were created using the OneHotEncoder, to reduce the dimensionality of the data. This will help reduce the model’s complexity and mitigate overfitting.

Using regular one-hot encoding without specifying the top categories end up returning too many binary variables. It also causes the data frame to become highly fragmented which causes poor performance and lead to overfitting in my case. Furthermore, it also caused my final R-squared value for x\_test to return a negative value. This meant that the data was an extremely poor fit for the model, and would not be useful for making predictions at all.

When testing other methods, I found that ordinal encoding returned a poorer RMSE and R-Squared results. This may be due to the fact that the ordinal encoding method is better suited for non-linear models. For other methods such as, count and frequency encoding, ordered ordinal encoding, target mean encoding and grouping rare/infrequent categories. I found that the RMSE and R-Squared results obtained from using one-hot encoding of frequent categories were much better than the results returned from the other methods.

I also tested ordered ordinal encoding after performing variable discretization, on numerical variables, to order the bins for a monotonic relationship between the bins and target. However, the RMSE and R-Squared results were better without ordering the bins. Therefore this method was not implanted in the final solution.

**Numerical Transformation**

Numerical transformation, also known as normalizing, involves using techniques to transform the numerical features to be on a similar scale. Numerous machine learning models tend to perform better when the numerical values have transformed and become normally distributed. Therefore, I have also implemented the use of normalization in my solution, to help improve the performance and training of the model.

The numerical transformation method used in my final solution is power transformation. This transformation was implemented into the solution through feature engine PowerTransformer function. For the function, lambda was specified as 1/3 as it returned the best transformation results of the exponents tested. The numerical transformation was done on the dataset itself as transformation had to be done on the target variable as well. This meant I had to perform the transformation before the train test split as I was unable to find a way to transform the target variable after the split.

I have tested other numerical transformation methods as well, and these were the results.

Log transformer could not be used as the features in the dataset contained zero and/or negative values. Log transformer can only be used on positive features, therefore, it was ruled out as an option.

Reciprocal function transformation also could not be used as the features in the dataset contained zero values which would cause the transformation to return an error.

Box-Cox transformation was also not applicable for the dataset. When trying to transform the data with the Box-Cox transformer function from feature engine, it would return an error message that stated “Data must be positive”. This leads me to believe that Box-Cox transformation can not be used for negative values. Similarly, Square / Cube root transformation could not be applied to the dataset for the same reason as Box-Cox.

The last transformation method tested was the YeoJohnson transformation method, which is an extension of the Box-Cox transformation method. This method was able to work on the dataset as the transformer was capable of handling negative and zero values. However, when comparing the RMSE and R-Squared results for both methods, I found that the power transformation method resulted in better model performance. Therefore, the method I chose to use in the end was the power transformer.

Apart from using power transformation on the data, I also implemented the use of discretization / binning. Discretization involves transforming continuous variables into discrete variables to minimize the influence of outliers and change the distribution of skewed variables. Discretization also improved the performance of the model. Equal-width discretization was the method used for the solution. Not only did it provide the best model performance over other methods, it was also more useful than equal-frequency discretization which is used more for skewed variables.

**Feature Engineering**

This section on feature engineering will cover the use of feature scaling as well as the creation or removal of features from the dataset. Then make comparisons to determine whether they improve the model performance and the method that is most suited for the model.

Feature scaling can be an important step for processing the data before modelling. It consists of standardization and normalization to scale the data. This ensures that the values in the features are not significant from each other, as significant values can impact the model’s performance due to their magnitude. This is why feature scaling is used, to reduce the distance between numbers and improve model performance, though that may not always be the case.

When comparing the model performance with and without feature engine scaling, I found that the model would perform better after applying some form of scaling on the datasets. This meant that feature scaling was a useful tool to improve model performance. The feature scaling method that was implemented in the final solution is the Maximum Absolute Scaler with centring. This method centres the distributions at zero and then scales it to its absolute maximum. This method uses two transformers. StandardScaler is used to remove the mean for centring, but does not divide by the standard deviation. Then the MaxAbsScaler is used to scale the distributions to its absolute maximum.

After performing feature scaling, a kernel density estimate plot is set up to compare the distribution of observations in the dataset before and after scaling. This allows us to observe the impact that the scaler had on the dataset. If you look at the graph shown in the coding section of the report, you can see that there is only slight changes in the shape of the graph, but the density and distribution of the variables has been scaled down.

The Maximum Absolute Scaler with centring returned the best model results based on the other results from the other feature scaling methods that I have tested. While standard and min-max scaler is the most preferred method for feature scaling in real-life applications, when I ran it for my dataset, the model performance when using these two scalers did not perform as well as the maximum absolute scaler. Mean normalization and Robust scaling also had similar results, meaning that they did not perform as well as the maximum absolute scaler in terms of model performance.

As for creating and removing features, I have removed one of the features from the dataset to help improve my model’s performance. This feature was “Outlet\_Establishment\_Year”. This feature was removed as based on my Exploratory Analysis conducted prior to data pre-processing, I have found that this feature has low correlation to the target variable and would not be helpful in making predictions. Removing this feature not only improved the performance and accuracy of the model, but also reduced the amount of overfitting present in the model.

**Linear Regression Model (explain, then show all results based on the steps taken to obtain the results)**

Finally, I will cover the linear regression model and the results that I have obtained from the model.

Linear regression is a model approach that is used to determine if an observed data set of values can be fitted onto a predictive model for prediction. In this report, the linear regression model has already been built beforehand. Therefore, I only had to input the data into the model, then examine and evaluate the results. The results can be evaluated by using the Root Mean Square Error (RMSE) as well as the R-Squared value.

RMSE measures the average of the squares of the errors, and it is used as a measure of the quality of a model. The lower the RMSE value, the better the model fits the dataset. On the other hand, R-Squared measures the proportion of the variance in a regression model. The closer the R-squared value is to 1, the better the model fits the dataset.

The results that I have obtained from the linear regression model based on the final combination of methods used is as follows,

train\_rmse: 3.55503628526283

test\_rmse: 3.8632842010772603

train\_r-sq: 0.7025075927698654

test\_r-sq: 0.6914339594063271.

These are the final and best results that I have obtained from the linear regression model after trial and error while testing the various methods to perform the data wrangling.

**Summary**

To summarise, data wrangling involves utilizing various different data wrangling and exploration techniques, to conduct data exploration, cleaning, preparation and transformation on a real-life data set. Exploratory Data Analysis is done before any pre-processing to understand the different relationships between the variables and identify those that are useful for making predictions. After that, we move on to data cleaning which consists of handling missing values as well as outliers to ensure that the model performs well. Data pre-preparation involves data transformation which consists of categorical encoding and numerical transformation/normalization. Feature scaling as well as creating new or removing redundant features is also done during pre-processing. After processing the data, we can then load it into the linear regression model to examine the model performance, and evaluate the results.

The methods I used for my solution is as follows,

* Imputation pipeline for missing values
* Censoring for handling outliers
* Power Transformer with lambda = 1/3 for numerical transformation
* One-hot encoding with top categories = 5 for categorical encoding
* Numerical Discretization with Equal-Width Discretiser
* Maximum Absolute Scaler with centring

Which gave me the following linear regression results,

- train\_rmse: 3.55503628526283

- test\_rmse: 3.8632842010772603

- train\_r-sq: 0.7025075927698654

- test\_r-sq: 0.6914339594063271.

Improvements can still be made to the model’s performance and results. The linear regression results can be further improved by trying other combinations of methods that may have given more satisfactory results, as I do not believe that I have tried every possible method combination when processing the data. Exploratory analysis can also be further improve by examining other variables in greater detail; instead of just focusing on the target variable.

This concludes my report on the data wrangling assignment 1.