FIT 5196 – Data Wrangling

Assignment 2

Task 3: Project Reflective Report

Group 130

Name: Chaitanya Tambolkar

Student ID: 34093117

Name: Fahmid Tawsif Khan Chowdhury

Student ID: 34121315

Table of Contents

1. Introduction

This report reflects on the feedback received during Week 10 and outlines the steps taken to address the key issues raised by our tutor. The primary focus was on improving the handling of outliers, validating the *is_expedited_delivery* feature, and restructuring the code for better data processing. We implemented linear regression models to detect and correct errors in the dataset, such as mispredictions in delivery charges. Additionally, we reordered the error-fixing process based on column dependencies, ensuring the integrity of each feature. These changes resulted in significant improvements in the model's accuracy, as reflected by a notable increase in the R² score. This report documents the feedback received, the corrections made, and suggestions for further improvements in future tasks.

2. Tutor's Feedback

The following feedback was provided by our tutor:

a. **Section Overview:** Explain the methodology and discuss the findings in the beginning of every section.

b. **Expedited Delivery Check**: Investigate potential ways to check if orders marked as expedited reflect appropriate delivery charges.

c. **Handling Outliers**: Address the outliers in the dataset by considering a multivariate approach like linear regression.

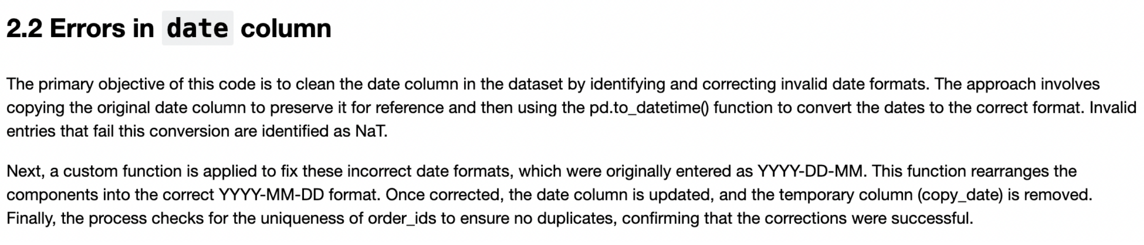
d. **Code Restructuring**: Prioritise which columns to fix in dirty data so that additional errors do not occur.

3. Reflection to Feedback

a. **Section Overview:** To address the feedback provided, we made the following adjustments to each section of the report. First, we introduced a brief explanation of the methodology at the beginning of every section, outlining the key steps taken during that phase of the analysis. This gives the reader clear insight into how the techniques were applied.

Following the methodology, we added a discussion of the main findings for each section. This highlights the most important results and their implications for the project. By implementing these changes, we aimed to improve the clarity and logical flow of the report, making it easier for the reader to follow the analysis and understand the significance of the results.

Evidence:

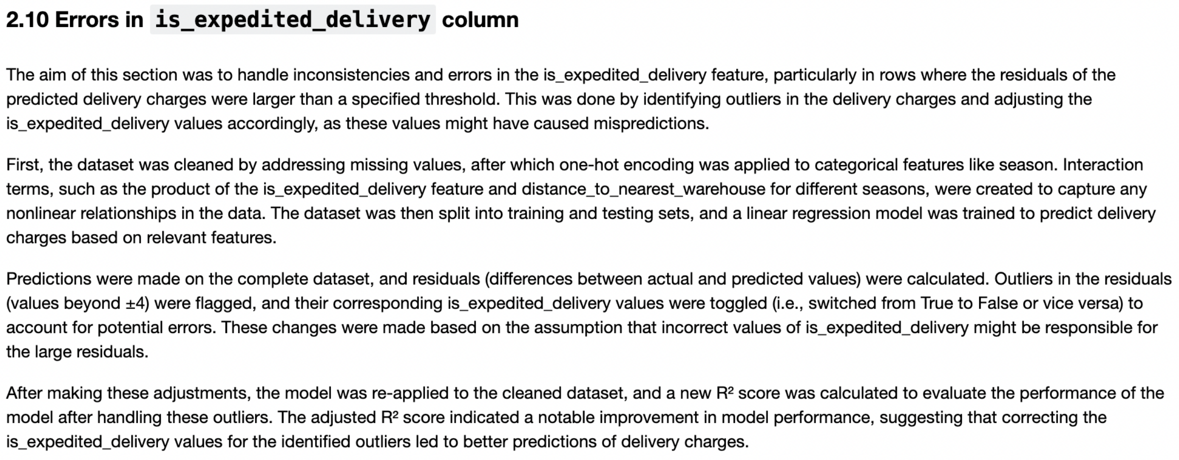


2.2 Errors in `date` column

The primary objective of this code is to clean the date column in the dataset by identifying and correcting invalid date formats. The approach involves copying the original date column to preserve it for reference and then using the pd.to_datetime() function to convert the dates to the correct format. Invalid entries that fail this conversion are identified as NaT.

Next, a custom function is applied to fix these incorrect date formats, which were originally entered as YYYY-DD-MM. This function rearranges the components into the correct YYYY-MM-DD format. Once corrected, the date column is updated, and the temporary column (copy_date) is removed. Finally, the process checks for the uniqueness of order_ids to ensure no duplicates, confirming that the corrections were successful.

Figure 1: Task 1 – fixing date errors in dirty data.



2.10 Errors in `is_expedited_delivery` column

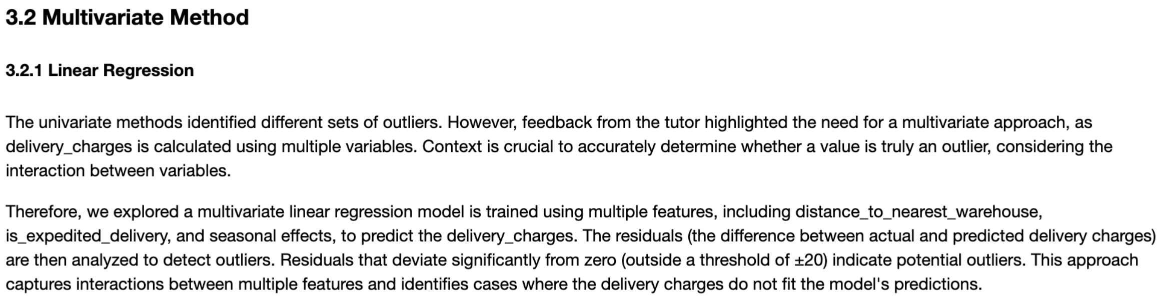
The aim of this section was to handle inconsistencies and errors in the is_expedited_delivery feature, particularly in rows where the residuals of the predicted delivery charges were larger than a specified threshold. This was done by identifying outliers in the delivery charges and adjusting the is_expedited_delivery values accordingly, as these values might have caused mispredictions.

First, the dataset was cleaned by addressing missing values, after which one-hot encoding was applied to categorical features like season. Interaction terms, such as the product of the is_expedited_delivery feature and distance_to_nearest_warehouse for different seasons, were created to capture any nonlinear relationships in the data. The dataset was then split into training and testing sets, and a linear regression model was trained to predict delivery charges based on relevant features.

Predictions were made on the complete dataset, and residuals (differences between actual and predicted values) were calculated. Outliers in the residuals (values beyond ±4) were flagged, and their corresponding is_expedited_delivery values were toggled (i.e., switched from True to False or vice versa) to account for potential errors. These changes were made based on the assumption that incorrect values of is_expedited_delivery might be responsible for the large residuals.

After making these adjustments, the model was re-applied to the cleaned dataset, and a new R^2 score was calculated to evaluate the performance of the model after handling these outliers. The adjusted R^2 score indicated a notable improvement in model performance, suggesting that correcting the is_expedited_delivery values for the identified outliers led to better predictions of delivery charges.

Figure 2: Task 1 - Fixing errors in is_expedited_delivery in dirty data.



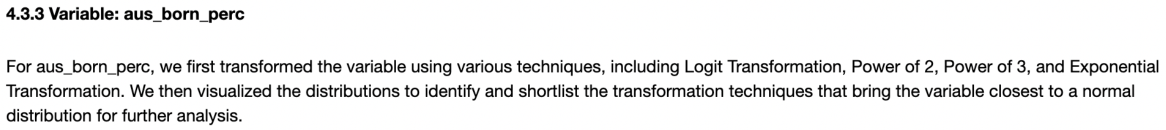
3.2 Multivariate Method

3.2.1 Linear Regression

The univariate methods identified different sets of outliers. However, feedback from the tutor highlighted the need for a multivariate approach, as delivery_charges is calculated using multiple variables. Context is crucial to accurately determine whether a value is truly an outlier, considering the interaction between variables.

Therefore, we explored a multivariate linear regression model is trained using multiple features, including distance_to_nearest_warehouse, is_expedited_delivery, and seasonal effects, to predict the delivery_charges. The residuals (the difference between actual and predicted delivery charges) are then analyzed to detect outliers. Residuals that deviate significantly from zero (outside a threshold of ±20) indicate potential outliers. This approach captures interactions between multiple features and identifies cases where the delivery charges do not fit the model's predictions.

Figure 3: Task 1 – Application of multivariate methods to detect outlier in outlier data.



4.3.3 Variable: aus_born_perc

For aus_born_perc, we first transformed the variable using various techniques, including Logit Transformation, Power of 2, Power of 3, and Exponential Transformation. We then visualized the distributions to identify and shortlist the transformation techniques that bring the variable closest to a normal distribution for further analysis.

Figure 4: Task 2 – Transformation of aus_born_perc.

b. **Checking for Expedited Delivery:** After receiving feedback, we implemented a check to validate the is_expedited_delivery column. We realised from the specification that the is_expedited_delivery was a variable influencing delivery_charges. Therefore, we modelled a linear regression using is_expedited_delivery to predict delivery_charges, and rows with large residuals from predicted delivery charges were identified as the rows with incorrect is_expedited_delivery.

Residuals between actual and predicted values were calculated, and outliers (residuals beyond ±4) were flagged. The *is_expedited_delivery* values for these outliers were toggled under the assumption that mispredictions could stem from incorrect entries. After reapplying the model to the cleaned dataset, the R^2 score improved from 0.8081 to 0.9447, indicating a significant enhancement in prediction accuracy, confirming the effectiveness of adjusting the *is_expedited_delivery* feature for outliers.

Evidence:

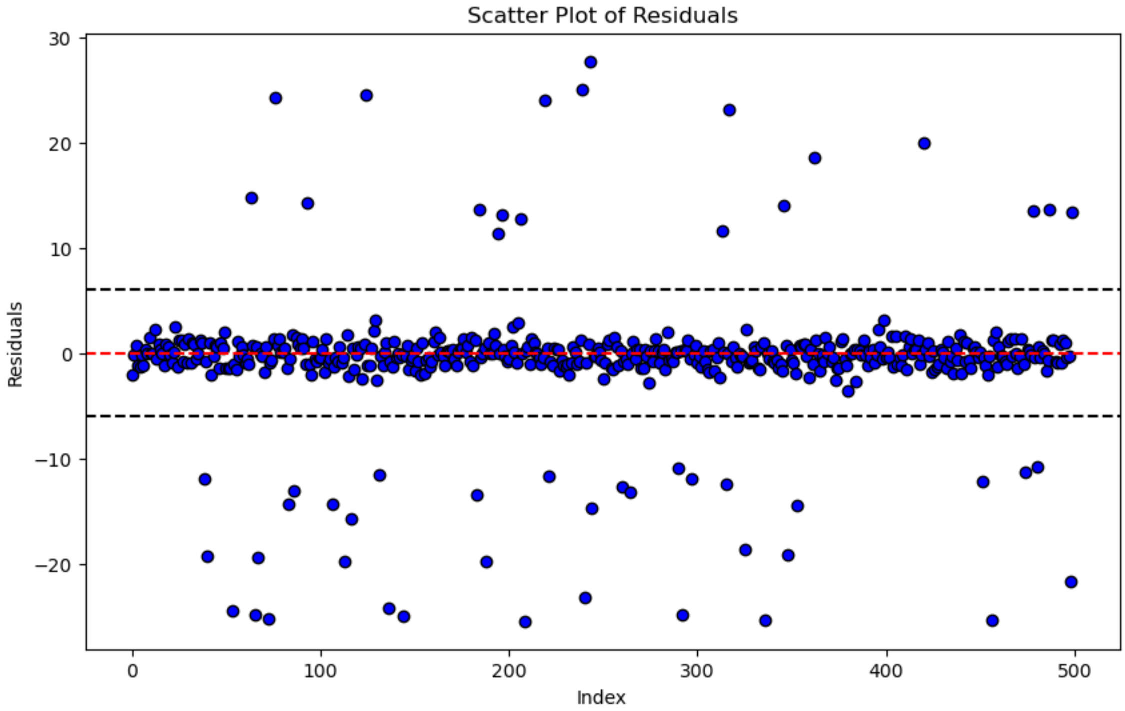


Figure 5: Task 1 – The residual plot helping to identify the outliers (is_expedited_delivery).

c. **Handling Outliers:** Before week 10, we initially used univariate methods (3sd rule, Hampel, boxplot) to handle outliers, which resulted in identifying different ranges of outliers. Based on our tutor's advice, we then applied a linear regression model with an R² value of 0.9956 to detect and address outliers more effectively. By using a residual plot, we identified boundaries of 20 and -20, successfully removing 20 outliers that fell outside these limits.

Evidence:

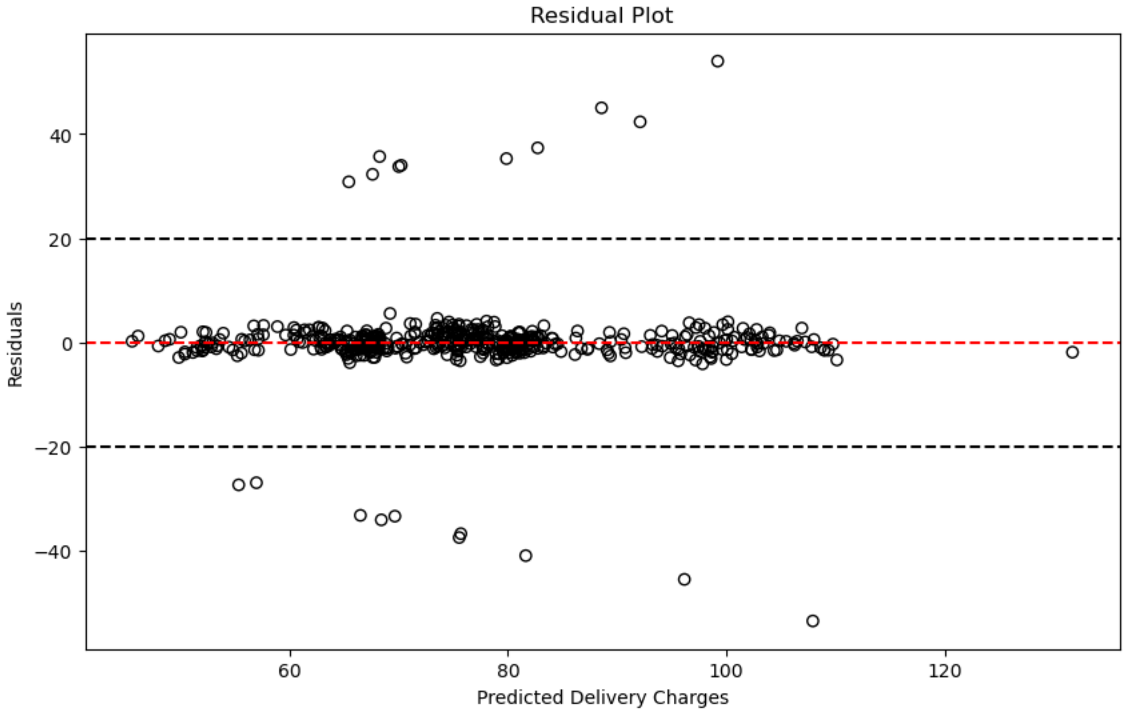
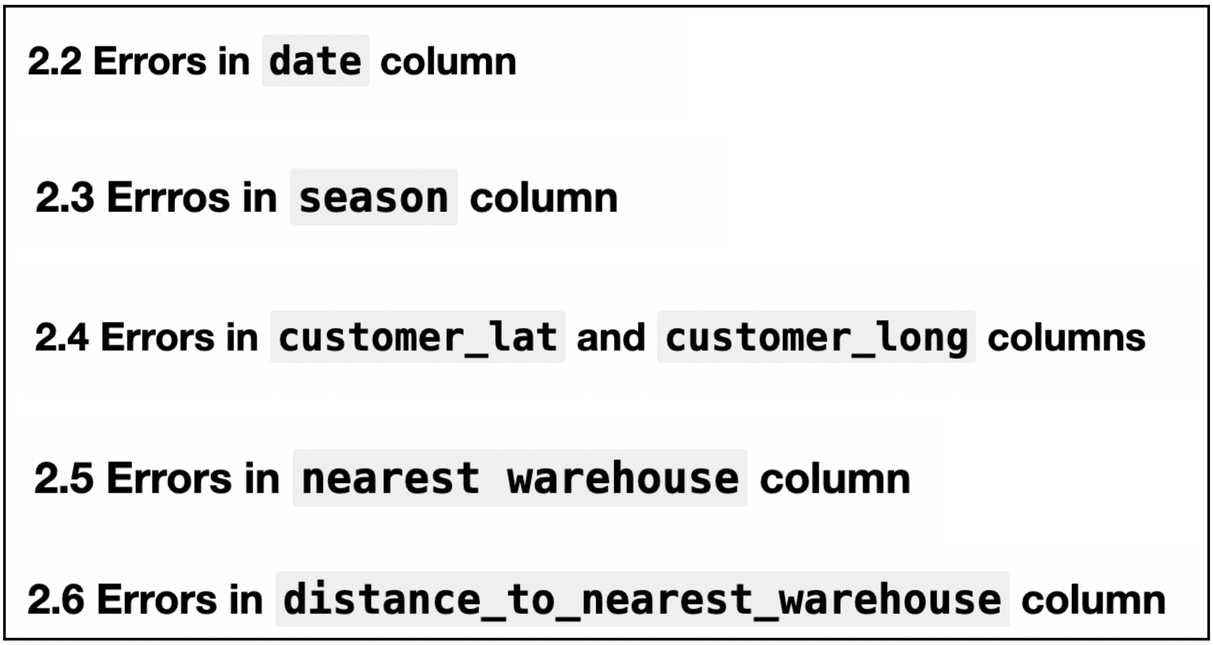


Figure 6: Task 1 – Outlier detection using a multivariate method – linear regression.

d. **Code Restructuring**: Based on the feedback, we prioritized error fixes by thoroughly analyzing the data. For instance, fixing the season column would have been incorrect without first addressing the date column, as season depends on the date. Similarly, we prioritized fixing order_total, which is directly dependent on order_price.

Evidence:



2.2 Errors in date column

2.3 Errros in season column

2.4 Errors in customer_lat and customer_long columns

2.5 Errors in nearest warehouse column

2.6 Errors in distance_to_nearest_warehouse column

Figure 7: Task 1 – Reorder Sections. Season depends on date, so date was prioritised. Similarly, distance_to_nearest_warehouse depends on nearest_warehouse, which in turn depends on customer_lat and customer_long; they were all apprpriately ordered to handle the errors properly.

4. Summary Table for Feedback and Action:

Feedback	**Reflection**	**Solution**
Methodology and findings should be explained at the beginning of every section.	We realized that providing context at the start of each section would improve clarity and flow for readers.	Introduced a clear explanation of the methodology and added discussions of key findings at the start of each section.
Investigate if *is_expedited_delivery* values correctly reflect appropriate delivery charges.	The *is_expedited_delivery* feature directly influenced *delivery_charges*, requiring validation for correct values.	Built a linear regression model to predict *delivery_charges*, identified outliers using residuals, and corrected errors.
Consider a multivariate approach like linear regression to address outliers.	Initial univariate methods identified outliers but didn’t handle all effectively. A multivariate approach was required.	Applied linear regression, identified outliers with residual boundaries of ±20, and successfully removed 20 outliers.
Prioritize error fixes in the correct order to avoid introducing additional errors.	We recognized that fixing dependent columns out of order could introduce new errors (e.g., *season* depends on *date*).	Prioritized fixing dependent columns first, starting with *date* for *season* and *order_price* for *order_total*.

5. Future Improvements

1. For task 1 - outliers, we could have applied KNN. However, we realised it too late and were unable to implement.
2. In task 2, while the applied transformations improved model performance, the process could be enhanced by addressing the presence of outliers that potentially impact the data distribution.
3. In Task 2, exploring interactions between variables could reveal more significant relationships not captured in individual transformations.

6. Conclusion

In conclusion, the feedback provided during Week 10 was instrumental in refining our approach to data handling, error correction, and model improvement. By applying multivariate methods like linear regression, we were able to detect and address outliers more effectively, improving the model's performance. The validation and adjustment of the *is_expedited_delivery* feature, along with a structured approach to resolving column dependencies, further enhanced the accuracy of our predictions. Though we identified areas for further improvement, such as applying KNN for outlier detection and exploring additional variable interactions, the steps taken in this project led to a significant increase in model performance, demonstrating the importance of iterative refinement based on feedback.