Name: Poojal Katiyar
Roll Number: 220770

1 Introduction

The purpose of this report is to analyze the Drift Diffusion Model (DDM) implemented to predict user purchase behavior based on various features of an e-commerce dataset. The model accumulates evidence over time until a decision threshold is reached, indicating whether a purchase occurs.

The dataset consists of 12,330 sessions, each representing a unique user. This approach helps avoid biases related to specific campaigns, special days, user profiles, or seasonal effects. The features included in the dataset provide insights into user interactions and engagement on the e-commerce platform.

2 Methodology

2.1 Data Cleaning

- Upon examining the dataset, we find that there are no missing values in any of the columns.
- Next, we transformed two features, Revenue and Weekend, which were originally in boolean format, into binary integers (0 and 1). This transformation makes them easier to use in subsequent calculations, particularly in machine learning models where binary variables are often more practical.

2.2 Data Analysis

- **Correlation Analysis** We begin by selecting only numeric features for correlation analysis. The correlation matrix is calculated and displayed using a heatmap:

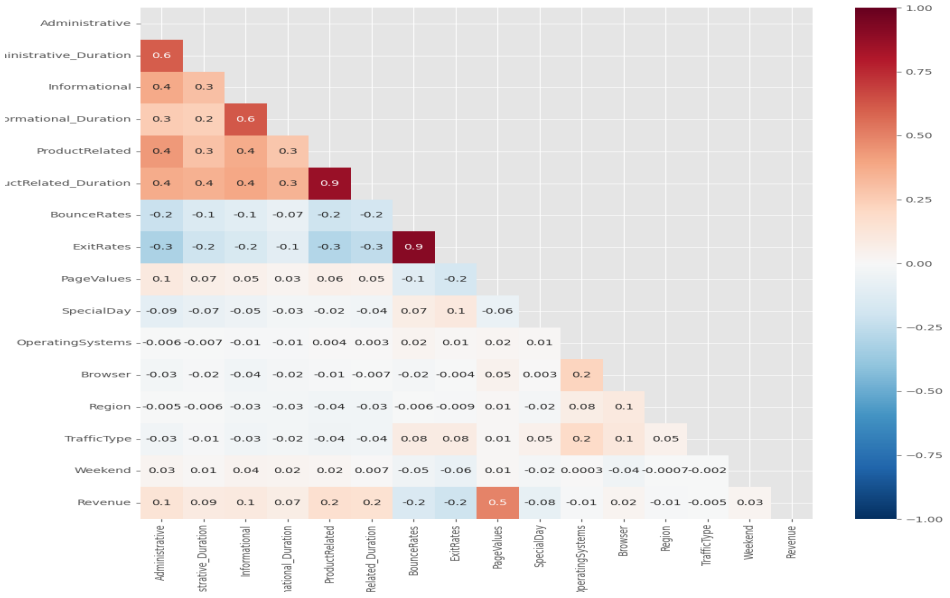

Figure 1: Heat Map

Name: Poojal Katiyar
Roll Number: 220770

- We can see is that there is very little correlation in general. The cases where correlation is high is between :
  - BounceRates & ExitRates (0.9)
  - ProductRelated & ProductRelated_Duration (0.9)
- Moderate Correlations: Among the features: Administrative, Administrative_Duration, Informational, Informational_Duration, ProductRelated, and ProductRelated_Duration.
- Revenue has a moderate correlation with Page Value.
- **PairPlot Analysis:** Next, we examine pairwise relationships between selected features and Revenue using a pairplot:

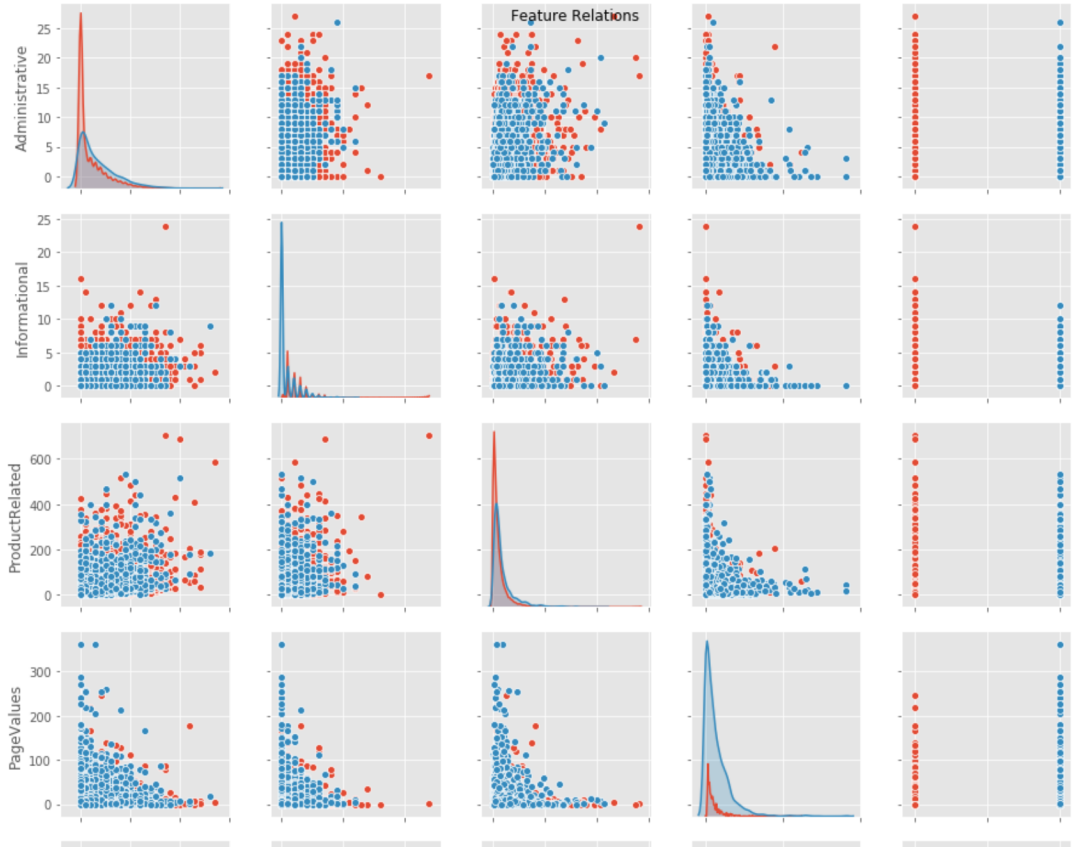

Figure 2: Pair Plot

- Revenue does not show strong correlation with any of the features, indicating that the relationship between these variables and the likelihood of a purchase (Revenue) is weak.
- **Violin and Box Plots for Various Features by Revenue: Observations:**
  - Visitors who do not make a purchase (Revenue = 0) tend to visit fewer pages and spend less time on the site.
  - Visitors who make a purchase tend to view more product-related pages and spend more time on them compared to informational or account-related pages.

- You can see in the following image:

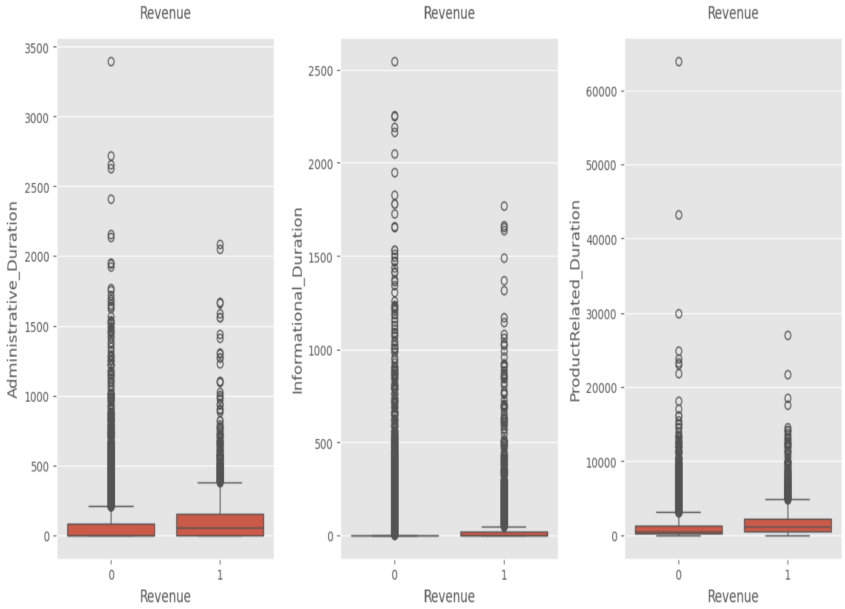

Figure 3: Box Plots

- **Page Metrics Analysis:**
  - All three plots show a large number of outliers.
  - ExitRates tend to have higher values than BounceRates. This is likely due to certain pages (like transaction confirmation pages) having high exit rates that influence the overall average.

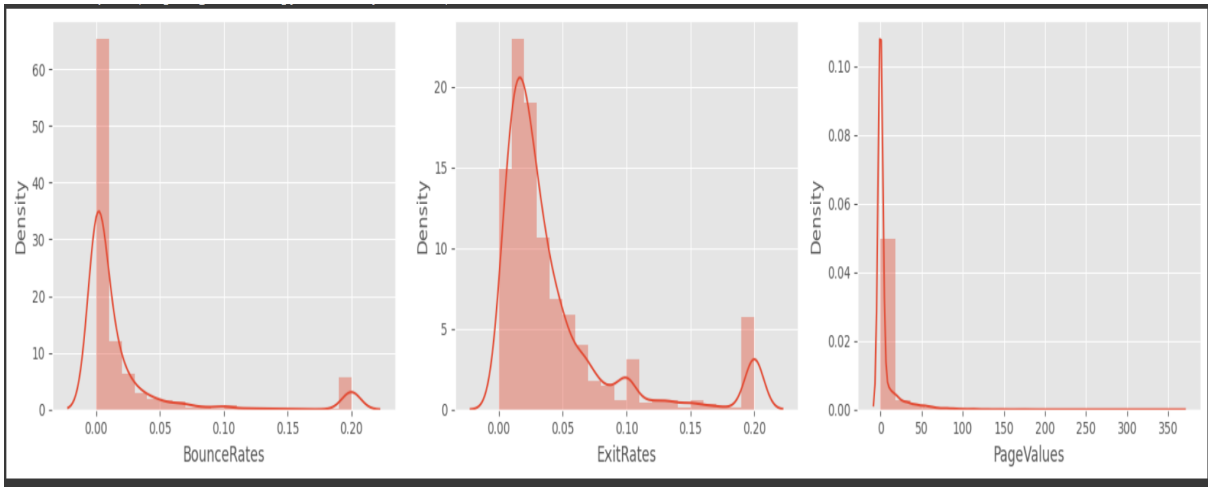

Figure 4: Page Metric Plot

- **Some more analysis and their plots:**
  - Most transactions happen on special days (SpecialDay =0).The closer the visit date is to a special day (like black Friday, new year, etc.). etc) the more likely it will end up in a transaction.

– Weekends probably don't impact too much on the transactions as seen from the graph. There is also lot of variation in the transaction happening in different months. Through graph, we founded that November has the highest purchases while January has the least.

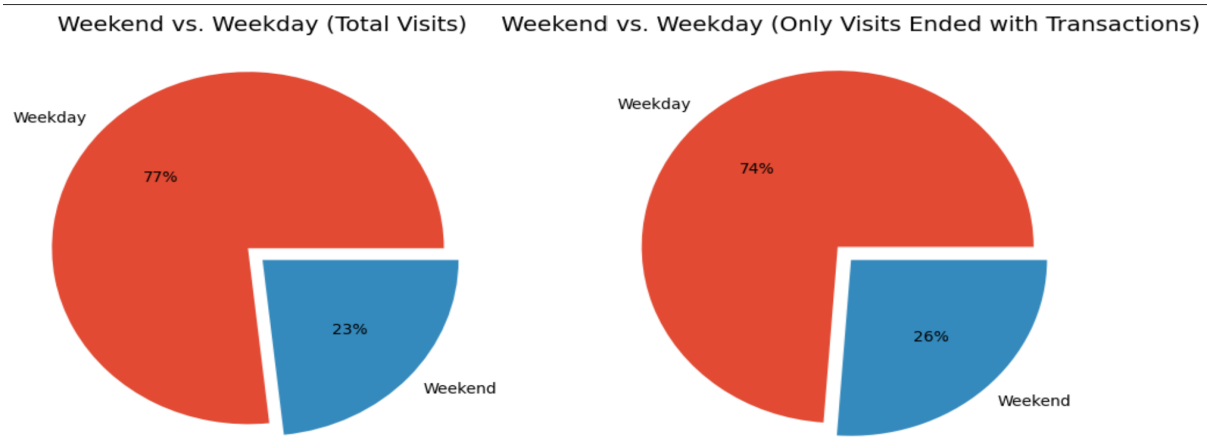

Figure 5: Weekdays vs Weekend

2.3 Data Preprocessing

- A new column named as Response Time is added in the existing table. This is the sum of three different kinds of duration which are Administrative, Product Related and Informative.
- After this I did one hot encoding of categorical variables like month and visitor type.
- I have made two new columns as drift rate and initial Bias for all rows. This drift rate will be the slope of the evidence Vs Time graph. The Initial Bias represents the starting point of evidence accumulation before the session begins.It will be different for each row. It will depend on parameters like Weekend, Special Day, Bounce Ratio, etc.
- Now I have grouped each session on the basis of Browser type into 13 products. Each browser type means new product session.

2.4 Modelling of Drift Rate and Initial Bias

- Firstly I applied linear regression with these parameters as input and revenue as output (1 if revenue=True otherwise 0). Then I got the weights corresponding to each parameter as well as a common bias. This I got for each product. Each product will have different threshold, common bias and weights corrsesponding to each parameter.
- Approximately I calculated the drift rate as the sum of the weights corresponding to Administrative, Product Related and Informative Duration. As these parameters vary with time, therefore they were considered. The weight assigned to each of

these durations determines their contribution to the drift rate (the speed at which the decision-making process occurs). As these durations change over time, they affect the drift rate, hence their inclusion in the model.

- The initial bias is calculated as the sum of the dot product of the weight and the respective values of the features that influence the decision-making process. These weights are of those parameters other than the above mentioned as they are constant and does not vary with time.Apart from the initial bias we have a common bias calculated from regression.

- **Snippet of the Code:**

```
for index, row in dff[dff['Browser'] == product_type].iterrows():
    # Calculate drift_rate for each row based on feature weights
    drift_rate = (feature_weights.get('Administrative_Duration', 0) +
                  feature_weights.get('Informational_Duration', 0) +
                  feature_weights.get('ProductRelated_Duration', 0))
    dff.at[index, 'drift_rate'] = drift_rate

    # Calculate initial_bias for each row based on feature weights
    A_columns = ['Administrative_Duration', 'Informational_Duration', 'ProductRelated_Duration', 'drift_rate', 'initial_bias']

    # Prepare Bias_weights by excluding the A_columns from the list of features
    Bias_weights = {feature: weight for feature, weight in feature_weights.items() if feature not in A_columns}

    # Ensure that the row has the right columns for Bias_weights (handle missing columns)
    row_features = [feature for feature in Bias_weights.keys() if feature in row.index]
    row_values = row[row_features].values

    if len(row_values) == len(Bias_weights):
        # Calculate the initial_bias using the dot product of the weights and the feature values
        initial_bias = np.dot(list(Bias_weights.values()), row_values)

        # Update the initial_bias value for the current row (add intercept to initial_bias)
        dff.at[index, 'initial_bias'] = initial_bias + bias  # Add the bias to the initial_bias
```

Figure 6: Code Snippet

- **E.g: Weight corresponding to Browser numbered 13**

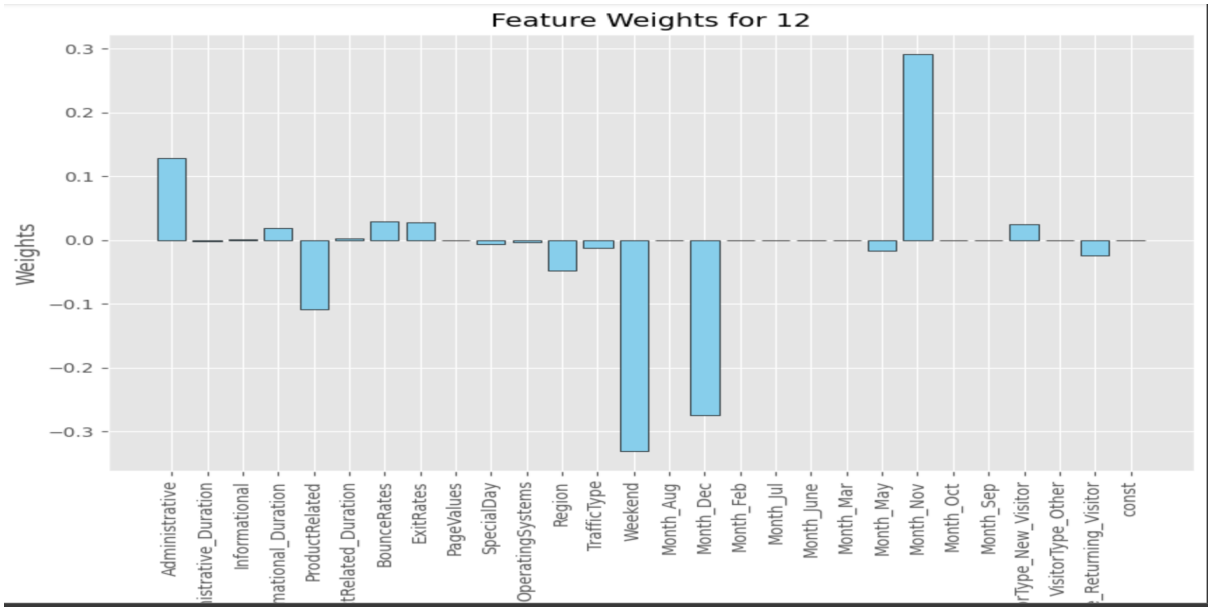

Figure 7: Weights

Name: Poojal Katiyar
Roll Number: 220770

2.5 Determining the threshold value for each product type:

- The threshold is determined using the Receiver Operating Characteristic (ROC) curve, a technique used in classification problems to evaluate model performance at various threshold settings. The true values (y_{true}) and the predicted probabilities (y_{pred}) from the linear regression model are used to compute the ROC curve. The `roc_curve()` function from `sklearn.metrics` returns:
  - **False Positive Rate (FPR)**: The proportion of negative instances incorrectly classified as positive.
  - **True Positive Rate (TPR)**: The proportion of positive instances correctly classified.
  - **Thresholds**: Different possible decision boundaries.

  The optimal threshold is selected based on the maximum difference between TPR and FPR. The formula used is:

$$\text{optimal\_index} = \arg\max(\text{TPR} - \text{FPR})$$

  This approach maximizes the true positive rate while minimizing the false positive rate, effectively balancing sensitivity and specificity.

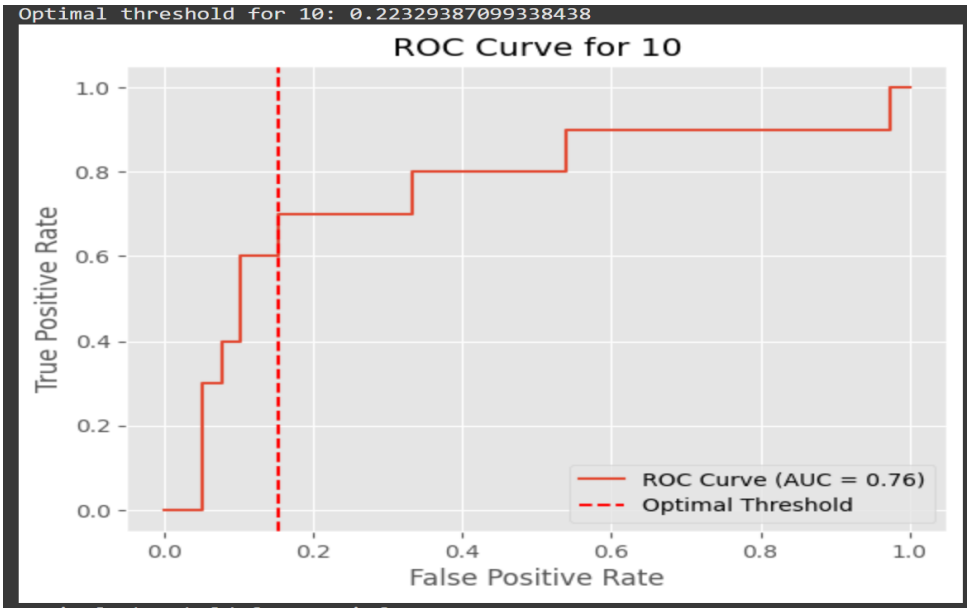

Figure 8: Threshold

2.6 Modelling the Drift Diffusion Model Curve using drift Rate a, bias and noise

- Here is the graph for Product type-2 where the purchase is made and the Evidence becomes greater than the threshold as seen in the figure:

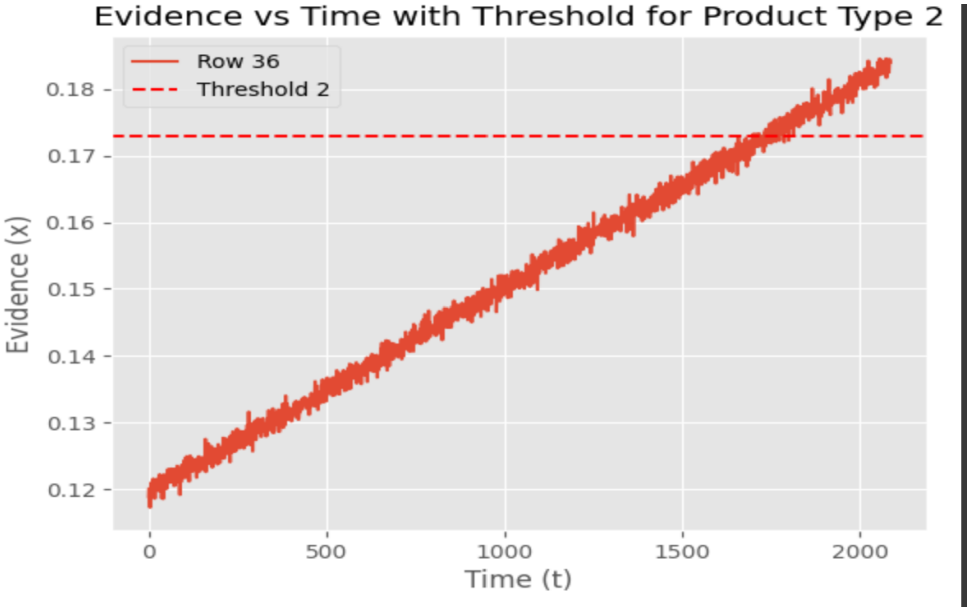

Figure 9: DDM Curve

2.7 Visualizations:

- Comparison of mean and standard deviation for different Product Type:

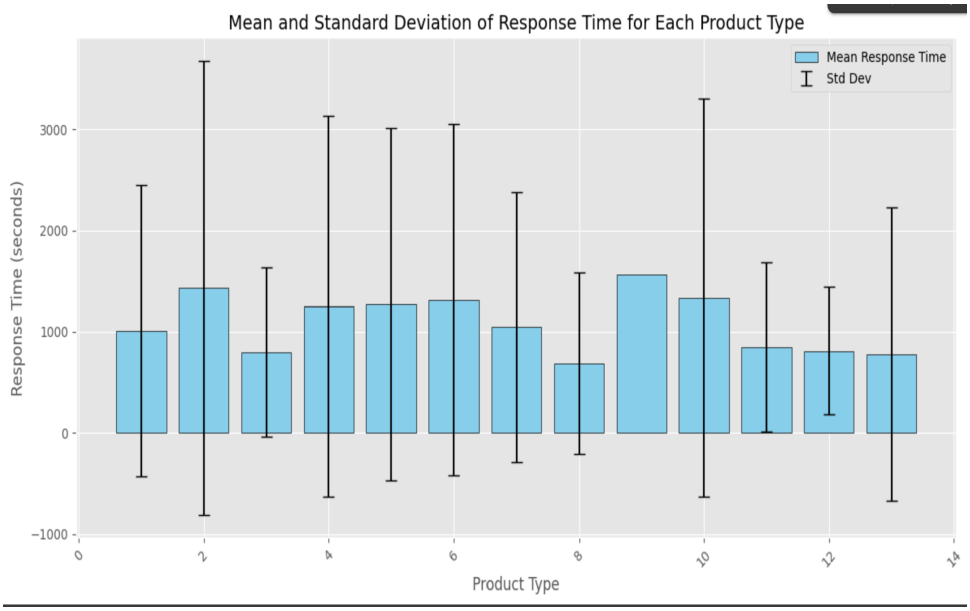

Figure 10: Mean and S.D. across products

- The drift values for different products are different. It can be negative as well as positive. According to the graph above we can conclude that Product 8 was the fastest purchased item.

- Drift Values corresponding to different products:

1	0.000110
2	0.000031
3	-0.000273
4	-0.000106
5	0.000145
6	-0.000059
7	0.001002
8	-0.368304
9	0.000000
10	0.000398
11	0.000519
12	0.018988
13	0.801765

Figure 11: Drift Values:

2.8 Interpretations:

- While the drift rate (a) plays a significant role in determining how quickly a decision is made, it is not the sole factor influencing response times. Negative drift values may indicate a delay in purchase decisions, where customers take longer to decide or may even avoid purchasing.Products with lower mean reaction times (faster decisions) likely represent products with stronger purchase intent, high appeal, or simpler decision-making. A higher standard deviation indicates greater variability in decision times. The following interpretations provide insight into why a product with the highest drift rate may not always lead to the quickest purchase decision:

- **Threshold Sensitivity:**

  - Even with a high drift rate, if the decision threshold is set too high or if there is significant noise, the decision may take longer to accumulate enough evidence. A product with a high drift rate might not always result in the fastest purchase if the threshold for decision-making is high.
  - **Interpretation:** A product with a lower decision threshold might result in quicker decisions, even if its drift rate is not the highest.

- **Bias in Evidence Accumulation:**

  - Bias in the decision process, such as a strong predisposition towards a product, can influence how quickly evidence accumulates. A product that has a positive bias may see quicker decisions, even if its drift rate is not the highest.
  - **Interpretation:** A product with a high positive bias may lead to faster purchases, even though its drift rate is lower than other products with higher cognitive engagement or deliberation.

- **Noise Variability:**

  - Noise in the decision process can influence how quickly evidence accumulates. High noise might delay the decision, even if the drift rate is high.

  - **Interpretation:** A product with high noise might result in slower decision-making, leading to longer response times despite having a high drift rate. Conversely, a product with lower noise might result in quicker decisions, even if the drift rate is moderate.

- **Complexity of the Product or Decision:**
  - Products with complex decision-making processes might take longer to purchase despite having a high drift rate. For example, a high-drift product might still require careful consideration or comparison with alternatives before the decision is finalized.
  - **Interpretation:** A product with a low drift rate but simpler decision-making processes might see quicker purchases because users do not deliberate for long before making a decision.

- **Overlapping Decision Boundaries (Threshold Reaching):**
  - In the case of multiple products, the drift rates may converge, causing decisions to be delayed despite high drift rates for some items. Products with a lower drift rate might hit the decision threshold more quickly due to smaller decision boundaries or less opposing noise.
  - **Interpretation:** Even if a product has a lower drift rate, it might hit the threshold earlier in certain conditions due to more stable evidence accumulation, such as less noise or better bias.

2.9 Conclusion

The Drift Diffusion Model has provided valuable insights into the dynamics of purchase decision-making. By analyzing response times and drift rates across different product types, I have been able to identify key factors influencing purchase behavior.

- **Colab Link for the Code:** https://colab.research.google.com/drive/1NFh0-uAKrLYH-oXjjqb5QTVEcWK4PNemscrollTo=A7ESdkaWR5Mp