Republic of the Philippines
North Eastern Mindanao State University
LIANGA CAMPUS
Lianga, Surigao del Sur

COLLEGE OF INFORMATION TECHNOLOGY EDUCATION

Customer Shopping Trends Dataset

Journey into Consumer Insights and Retail Evolution with Synthetic Data

A Final Requirement for the subject

Automata Theory and Formal Language

Submitted by:

DEXTER L. ALOLOD

Table of Contents

Abstract

Customer Shopping Trends Dataset is a comprehensive synthetic dataset created for educational purposes in the field of data analysis and machine learning. The dataset, containing 3900 instances, encapsulates a wide array of customer attributes, including age, gender, purchase history, preferred payment methods, shopping frequency, and more. The objective is to glean insights into consumer behavior and purchasing patterns, which are crucial for businesses to customize their product offerings, marketing strategies, and overall customer experience.

The research involves testing and loading the Customer Shopping Trends Dataset into five different classification techniques: BayesNet, Simple Logistic, Random Forest, Decision Table, and KStar. These methods were selected for their ability to handle diverse data features and provide insights from various analytical perspectives. The dataset's comprehensive nature, with features like purchase amount, feedback ratings, type of items purchased, and interaction with promotional offers, makes it an ideal candidate for such an extensive analysis.

By applying these classification techniques, the study aims to uncover patterns and trends within the data, thereby enabling businesses to make data-driven decisions. This analysis is expected to contribute to the optimization of product offerings and enhancement of customer satisfaction by aligning business strategies with customer needs and preferences. The findings of this study are anticipated to provide valuable insights for beginners in data analysis and machine learning, offering a practical demonstration of how these techniques can be applied to real-world datasets.

BayesNet

```
Scheme:       weka.classifiers.bayes.BayesNet -D -Q weka.classifiers.bayes.net.search.local.K2 -- -P 1 -S BAYES -E weka.classifiers.bayes.net.estimate.Simple
Relation:     shopping_trends_updated
Instances:    3900
Attributes:   18
              Customer ID
              Age
              Gender
              Item Purchased
              Category
              Purchase Amount (USD)
              Location
              Size
              Color
              Season
              Review Rating
              Subscription Status
              Shipping Type
              Discount Applied
              Promo Code Used
              Previous Purchases
              Payment Method
              Frequency of Purchases
Test mode:    5-fold cross-validation
```

Figure 1: BayesNet model run Details for Shopping Trends Analysis

The Bayesian Network implemented in Weka forms the core of the machine learning model, utilizing the K2 algorithm for structure learning, as indicated by its parameters (e.g., -P 1 -S BAYES). The model employs the SimpleEstimator for probability estimation, with an alpha parameter set at 0.5. The dataset in use, named "shopping_trends_updated," presumably contains data regarding shopping trends and comprises 3,900 entries or records. This dataset is characterized by 18 attributes, including Customer ID, Age, Gender, Item Purchased, Category, Purchase Amount (USD), Location, Size, Color, Season, Review Rating, Subscription Status, Shipping Type, Discount Applied, Promo Code Used, Previous Purchases, Payment Method, and Frequency of Purchases. Each attribute provides specific insights into the shopping behavior captured in the data. The evaluation of this dataset employs a 5-fold cross-validation method, an effective approach for testing how well the model generalizes to independent datasets. This method involves dividing the data into five parts, training the model on four parts while testing it on the fifth, and repeating this process five times to ensure each part is used once as the test set.

```
=== Classifier model (full training set) ===

Bayes Network Classifier
not using ADTree
#attributes=18 #classindex=17
Network structure (nodes followed by parents)
Customer ID(1): Frequency of Purchases
Age(1): Frequency of Purchases
Gender(2): Frequency of Purchases
Item Purchased(25): Frequency of Purchases
Category(4): Frequency of Purchases
Purchase Amount (USD)(1): Frequency of Purchases
Location(50): Frequency of Purchases
Size(4): Frequency of Purchases
Color(25): Frequency of Purchases
Season(4): Frequency of Purchases
Review Rating(1): Frequency of Purchases
Subscription Status(2): Frequency of Purchases
Shipping Type(6): Frequency of Purchases
Discount Applied(2): Frequency of Purchases
Promo Code Used(2): Frequency of Purchases
Previous Purchases(1): Frequency of Purchases
Payment Method(6): Frequency of Purchases
Frequency of Purchases(7):
LogScore Bayes: -88451.04376247838
LogScore BDeu: -93023.20096575306
LogScore MDL: -91586.18540530777
LogScore ENTROPY: -88088.51184032198
LogScore AIC: -88934.51184032198
```

```
Time taken to build model: 0.32 seconds
```

Figure 2: BayesNet Classifier model (full training set)

Based on the provided figure, the machine learning model is a Bayes Network Classifier predict customer behavior, with a focus on the frequency of their purchases. At the heart of this model is a probabilistic approach based on Bayes' theorem, which is adept at calculating the likelihood of various outcomes from a range of input features. Key to the model's functionality are its 18 different attributes or features. The most critical among these is the target variable, 'Frequency of Purchases', which is the 18th attribute (indexed as 17, considering the starting index is 0). The network structure of the model is intriguing, as it showcases dependencies among attributes. For example, the 'Customer ID' attribute is dependent on the 'Frequency of Purchases' attribute, suggesting a complex interplay of customer identification with their purchasing frequency.

A detailed look at the attributes reveals a comprehensive list that includes 'Customer ID', 'Age', 'Gender', 'Item Purchased', 'Category', and more. Each attribute sheds light on different aspects of the customer or their purchase. Intriguingly, some attributes, like 'Item Purchased(25)', might signify the presence of multiple distinct categories or levels within that attribute. The model's performance is gauged using various LogScores - including Bayes, BDeu, MDL, ENTROPY, and AIC. These scores are critical metrics in evaluating how well the model fits the data, with lower scores generally indicating a more accurate model.

Remarkably, the model boasts of high efficiency, taking only 0.32 seconds to build. This quick processing time underscores the effectiveness of the algorithm used in handling the data. In essence, this dataset serves as a tool to decipher the relationship between diverse factors such as age, gender, and type of item purchased, and their impact on how frequently customers make purchases. The Bayes Network Classifier is at the forefront of this endeavor, aiming to accurately predict purchase frequency based on these varied attributes.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         588               15.0769 %
Incorrectly Classified Instances      3312               84.9231 %
Kappa statistic                          0.009
Mean absolute error                      0.2451
Root mean squared error                  0.3564
Relative absolute error                100.0958 %
Root relative squared error            101.8511 %
Total Number of Instances             3900
```

Figure 3: Stratified cross validation of BayesNet

The evaluation of a machine learning model using stratified cross-validation revealed several critical insights. Firstly, the model's accuracy is relatively low, with only 15.08% (588 out of 3900 instances) of the instances being correctly classified. This is complemented by a high rate of incorrectly classified instances at 84.92% (3312 instances), pointing to potential issues in model performance. The Kappa Statistic, at a mere 0.009, indicates that the agreement between predicted and actual classifications is almost random, suggesting little to no predictive reliability.

Furthermore, the model exhibits moderate prediction error, as indicated by a Mean Absolute Error of 0.2451. This is further supported by a Root Mean Squared Error of 0.3564, which emphasizes the presence of significant prediction errors, particularly giving more weight to larger discrepancies. The Relative Absolute Error, standing at an alarming 100.0958%, and the Root Relative Squared Error at 101.8511%, both exceed 100%, indicating that the model's predictions are generally worse than those of a naive model that simply predicts the average.

Finally, the total number of instances used in this evaluation was 3900. Considering these metrics, it becomes evident that the model's performance is quite unsatisfactory, with low accuracy and high error rates. Based on these metrics, the model's performance seems quite poor, with low accuracy and high error rates. It's important to investigate further to understand why the model is not performing well, which might include examining the features used, the model's complexity, the quality of the data, or whether the problem is suitably framed for machine learning.

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.118	0.137	0.122	0.118	0.120	-0.019	0.490	0.136	Fortnightly
	0.124	0.117	0.146	0.124	0.134	0.008	0.471	0.131	Weekly
	0.168	0.147	0.164	0.168	0.166	0.020	0.490	0.148	Annually
	0.178	0.148	0.169	0.178	0.173	0.029	0.516	0.158	Quarterly
	0.166	0.150	0.153	0.166	0.160	0.016	0.496	0.139	Bi-Weekly
	0.170	0.141	0.166	0.170	0.168	0.028	0.515	0.150	Monthly
	0.130	0.150	0.132	0.130	0.131	-0.020	0.480	0.140	Every 3 Months
Weighted Avg.	0.151	0.142	0.150	0.151	0.150	0.009	0.494	0.143	

Figure 4: BayesNet Detailed Accuracy by Class

Based on figure 4, the model's effectiveness is assessed across various classes, each evaluated through several key metrics. Firstly, the TP Rate, or True Positive Rate, reflects the proportion of actual positive cases correctly identified by the model. For instance, in the 'Fortnightly' class, the model correctly identified 11.8% of the cases. Conversely, the FP Rate, or False Positive Rate, indicates the proportion of actual negative cases wrongly identified as positive, such as the 13.7% of non-Fortnightly cases incorrectly labeled as 'Fortnightly'.

Precision is another critical metric, measuring the proportion of predicted positive cases that were indeed positive. For the 'Fortnightly' class, a precision of 0.122 signifies that approximately 12.2% of the predictions made for 'Fortnightly' were correct. Recall, synonymous with the True Positive Rate, also reflects this aspect of model accuracy. The F-Measure, representing the harmonic mean of precision and recall, provides a singular score to encapsulate the model's accuracy. In the case of 'Fortnightly', the F-Measure stands at 0.120. Another important metric is the MCC, or Matthews Correlation Coefficient, which evaluates the quality of binary classifications, adapted for multi-class scenarios. It considers both true and false positives and negatives, offering a balanced measure irrespective of class size variations.

Additionally, the ROC Area, or the Area under the Receiver Operating Characteristic Curve, summarizes the model's ability to distinguish between positive and negative classes. An ideal model would have an area of 1, while an area of 0.5 suggests no discriminative power. The PRC Area, or Area under the Precision-Recall Curve, illustrates the balance between precision and recall at various thresholds, where a higher area indicates both high recall and precision. Each class, such as 'Fortnightly', 'Weekly', etc., is assessed using these metrics. The 'Weighted Avg.' row presents an average of these metrics across all classes, weighted by the number of instances in each class, providing a comprehensive view of the model's overall performance. Overall, the metrics suggest moderate to low performance across the classes, with none of the classes showing particularly high scores in precision, recall, or F-measure.

```
=== Confusion Matrix ===

  a   b   c   d   e   f   g   <-- classified as
 64  62  78  83  75  76 104 |   a = Fortnightly
 80  67  72  81  92  78  69 |   b = Weekly
 76  67  96  92  71  82  88 |   c = Annually
 73  63  86 100  92  78  71 |   d = Quarterly
 74  70  79  85  91  74  74 |   e = Bi-Weekly
 71  61  84  77  73  94  93 |   f = Monthly
 87  70  91  75 100  85  76 |   g = Every 3 Months
```

Figure 5: BayesNet Confusion Matrix

Figure 5 details the structure of a matrix used to represent class predictions versus actual classifications. In this matrix, each row symbolizes the instances that the model predicted to be in a specific class. For example, the first row displays how many instances were labeled as 'Fortnightly' by the model. Conversely, each column represents the actual class of the instances, with the first column showing the true 'Fortnightly' instances. The matrix entries, or cells, reveal the count of instances for a predicted class versus an actual class; the cell at the intersection of the first row and column, showing 64, indicates that 64 instances were accurately predicted as 'Fortnightly'. However, this row also displays misclassifications, such as 62 instances being incorrectly predicted as 'Fortnightly' when they were in fact 'Weekly', and so on.

The diagonal values, running from the top left to the bottom right of the matrix, represent true positives, where the number of correctly classified instances for each class is displayed, like 64 for 'Fortnightly', 67 for 'Weekly', and 96 for 'Annually'. Any value not on this diagonal signifies a misclassification. For example, the 62 in the first row but second column indicates 62 instances were wrongly classified as 'Fortnightly' instead of 'Weekly'. Lastly, the class labels at the end of each row denote the class that the row represents, such as 'a = Fortnightly', 'b = Weekly', and so forth. From this confusion matrix, we can analyze how well the model is performing, including which classes are being confused with each other. For example, high off-diagonal values indicate classes that are commonly misclassified as each other.

Simple Logistic

```
=== Run information ===

Scheme:       weka.classifiers.functions.SimpleLogistic -I 0 -M 500 -H 50 -W 0.0
Relation:     shopping_trends_updated
Instances:    3900
Attributes:   18
              Customer ID
              Age
              Gender
              Item Purchased
              Category
              Purchase Amount (USD)
              Location
              Size
              Color
              Season
              Review Rating
              Subscription Status
              Shipping Type
              Discount Applied
              Promo Code Used
              Previous Purchases
              Payment Method
              Frequency of Purchases
Test mode:    7-fold cross-validation
```

Figure 6: Simple Logistic Run Information

Figure 6 employs run information of the SimpleLogistic model, a logistic regression algorithm, configured with parameters (-I 0 -M 500 -H 50 -W 0.0). The dataset comprises 3,900 instances, each characterized by 18 attributes. These attributes cover a range of details, including Customer ID (a unique identifier), Age, Gender, Item Purchased, Category, Purchase Amount in USD, Location, Size, Color, Season, Review Rating, Subscription Status, Shipping Type, Discount Applied, Promo Code Used, Previous Purchases, Payment Method, and Frequency of Purchases. Furthermore, the dataset is analyzed using 7-fold cross-validation, a robust method to estimate the efficacy of the model in predicting outcomes on an independent dataset. The nature and diversity of these attributes suggest that this dataset is adept for modeling customer behavior, preferences, and the likelihood of purchasing specific items, with a particular emphasis on individual purchase details, customer demographics, and buying patterns.

```
=== Classifier model (full training set) ===

SimpleLogistic:

Class Fortnightly :
-0.04 +
[Location=Alaska] * 0.6

Class Weekly :
-0.04 +
[Location=Louisiana] * 0.55

Class Annually :
0    +
[Color=Magenta] * 0.5

Class Quarterly :
-0.01 +
[Location=Missouri] * 0.78

Class Bi-Weekly :
-0.03 +
[Color=Blue] * 0.44

Class Monthly :
0.01 +
[Item Purchased=Boots] * -0.58

Class Every 3 Months :
0.08 +
[Shipping Type=Standard] * -0.25
```

```
Time taken to build model: 6.66 seconds
```

Figure 7: Simple Logistic full training set

Figure 7 details the structured to classify purchase frequencies into categories like Fortnightly, Weekly, Annually, and others, using a SimpleLogistic logistic regression model. It operates by assigning classes based on specific attributes and their assigned weights. For instance, a 'Fortnightly' purchase frequency is predicted if the 'Location' attribute is 'Alaska', with a coefficient of 0.6. Similarly, 'Weekly' purchases are linked to 'Louisiana' with a 0.55 weight, while 'Annually' is associated with the color 'Magenta' and a 0.5 weight. 'Quarterly' purchases are tied to 'Missouri' with a 0.78 weight, and 'Bi-Weekly' to the color 'Blue' with a 0.44 weight. Intriguingly, a 'Monthly' purchase frequency is negatively correlated (weight of -0.58) with buying 'Boots', and purchasing every 3 months is negatively influenced (weight of -0.25) by 'Standard' shipping type. The model employs these coefficients for each attribute to calculate the likelihood of a purchase frequency class, with negative coefficients indicating a negative correlation. The model's build time was 6.66 seconds, reflecting its computational efficiency. Overall, this model appears adept at predicting the frequency of purchases based on various factors like location, item color, and other attributes, with coefficients indicating the strength and direction of each attribute's impact.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         574               14.7179 %
Incorrectly Classified Instances      3326               85.2821 %
Kappa statistic                          0.0011
Mean absolute error                      0.2448
Root mean squared error                  0.3503
Relative absolute error                 99.961  %
Root relative squared error            100.1062 %
Total Number of Instances             3900
```

Figure 8: Simple Logistic Stratified cross-validation

The dataset summary offers a comprehensive evaluation of a model's effectiveness through stratified cross-validation, focusing on its accuracy in predicting or classifying instances. A detailed examination reveals that only 574 instances, constituting 14.7179%, were correctly classified, while a significant 3326 instances (85.2821%) were incorrectly classified, indicating a substantial error rate. The Kappa Statistic, at a minimal value of 0.0011, implies negligible agreement between the model's predictions and the actual classifications, suggesting performance barely above chance level. In terms of error metrics, the Mean Absolute Error (MAE) stands at 0.2448, representing the average magnitude of prediction errors without directional consideration. The Root Mean Squared Error (RMSE) is 0.3503, highlighting the square root of the mean squared prediction errors and is particularly sensitive to larger errors. The Relative Absolute Error is alarmingly high at 99.961%, indicating that the model's predictions are nearly as imprecise as those of a naïve model predicting the mean outcome without any input variables. Correspondingly, the Root Relative Squared Error is also significantly high at 100.1062%, underscoring the model's inadequate performance. With the dataset encompassing 3900 instances for the cross-validation process, these metrics collectively point to the model's poor performance, evidenced by high error rates and a near-zero Kappa statistic.

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.046	0.054	0.122	0.046	0.067	-0.012	0.486	0.136	Fortnightly
	0.065	0.062	0.143	0.065	0.089	0.004	0.497	0.137	Weekly
	0.126	0.136	0.137	0.126	0.131	-0.011	0.499	0.149	Annually
	0.167	0.160	0.150	0.167	0.158	0.007	0.505	0.153	Quarterly
	0.048	0.045	0.148	0.048	0.072	0.005	0.495	0.140	Bi-Weekly
	0.071	0.066	0.150	0.071	0.096	0.006	0.510	0.147	Monthly
	0.485	0.476	0.152	0.485	0.231	0.006	0.515	0.152	Every 3 Months
Weighted Avg.	0.147	0.146	0.143	0.147	0.122	0.001	0.501	0.145	

Figure 9: Simple Logistics Detailed Accuracy by Class

The dataset provides a detailed accuracy of a classification model's performance evaluated across various classes like 'Fortnightly', 'Weekly', 'Annually', etc. Each class's performance is detailed through different metrics. The True Positive Rate (TP Rate) shows the proportion of actual positives that the model correctly identifies, exemplified by the TP rate of 0.046 for the 'Fortnightly' class, meaning it correctly identified 4.6% of such instances. In contrast, the False Positive Rate (FP Rate) indicates the proportion of actual negatives wrongly classified as positives; for 'Fortnightly', this rate is 0.054, indicating a 5.4% error in classification. Precision in this context is 0.122 for 'Fortnightly', signifying that only 12.2% of instances classified under this label were correct.

Recall, synonymous with TP rate, further measures the model's accuracy in identifying actual positives. The F-Measure, a combined metric of precision and recall, is at 0.067 for 'Fortnightly', providing a singular score of the model's accuracy. The Matthews Correlation Coefficient (MCC) offers a holistic measure of binary classifications, considering true and false positives and negatives, important for classes of varying sizes. The Receiver Operating Characteristic (ROC) Area evaluates the classifier's ability to differentiate between classes, with a score close to 0.5, as seen across all classes, indicating poor performance. Similarly, the Precision-Recall Curve (PRC) Area, which measures both precision and recall, further underscores the model's underperformance. The 'Class' column specifically names each class being evaluated.

Overall, the dataset reveals that the model struggles significantly across all classes, with low scores in TP Rate, Precision, and F-Measure. The 'Every 3 Months' class shows the highest TP Rate, but it's still below 50%. The consistent ROC Area of around 0.5 across all classes further highlights the model's inability to effectively distinguish between different classes.

```
=== Confusion Matrix ===

  a   b   c   d   e   f   g   <-- classified as
 25  36  69  82  24  50 256 |   a = Fortnightly
 32  35  71  84  22  29 266 |   b = Weekly
 34  31  72  92  19  41 283 |   c = Annually
 29  32  88  94  30  32 258 |   d = Quarterly
 24  33  70 103  26  38 253 |   e = Bi-Weekly
 32  40  70  89  21  39 262 |   f = Monthly
 29  37  86  84  34  31 283 |   g = Every 3 Months
```

Figure 10:Simple Logistic Confusion Matrix

Figure 10's table, often employed to describe the confusion matrix or the performance of a classification model, it provides a detailed breakdown of how the model classifies different instances. In this matrix, each row represents the instances in an actual class, while each column corresponds to instances in a predicted class. The labels (a, b, c, d, e, f, g) denote various classes: 'a' for Fortnightly, 'b' for Weekly, 'c' for Annually, 'd' for Quarterly, 'e' for Bi-Weekly, 'f' for Monthly, and 'g' for Every 3 Months.

The entries within the matrix reveal the count of instances for each combination of predicted versus actual class. For instance, a figure like 25 in the first row and first column signifies that the model correctly predicted 25 instances as 'Fortnightly', marking them as true positives. Conversely, a number like 36 in the first row, second column, indicates 36 instances where the model erroneously predicted 'Weekly' for actual 'Fortnightly' cases.

This matrix is instrumental in understanding not only the count of correctly classified instances (visible along the diagonal from the top-left to bottom-right) but also in pinpointing the types of errors the model is prone to. A notable number in the off-diagonal cells suggests a higher frequency of misclassifications. For example, the first row, representing 'Fortnightly', shows a significant number of instances misclassified across various classes, particularly 'Every 3 Months' with 256 instances.

Random Forest

```
=== Run information ===

Scheme:       weka.classifiers.trees.RandomForest -P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
Relation:     shopping_trends_updated
Instances:    3900
Attributes:   18
              Customer ID
              Age
              Gender
              Item Purchased
              Category
              Purchase Amount (USD)
              Location
              Size
              Color
              Season
              Review Rating
              Subscription Status
              Shipping Type
              Discount Applied
              Promo Code Used
              Previous Purchases
              Payment Method
              Frequency of Purchases
Test mode:    10-fold cross-validation
```

Figure 11: Random Forest run information

Figure 11 utilizes a RandomForest model run information. The behavior of the algorithm is tailored through specific parameters like -P 100 -I 100. It comprises 3900 instances, each encompassing 18 distinct features. These features include a unique Customer ID, Age, Gender, details of the Item Purchased, its Category, Purchase Amount in USD, the Location of purchase or customer, Size and Color of the item, Season of purchase, Review Rating, Subscription Status, Shipping Type, whether a Discount was Applied, usage of a Promo Code, the number of Previous Purchases, Payment Method, and the Frequency of Purchases.

Additionally, the dataset undergoes a 10-fold cross-validation, a robust method to gauge how the statistical analysis results might translate to an independent dataset. The configuration and feature set of this dataset suggest that the RandomForest model is possibly designed to predict or classify one of these attributes based on the others.

```
=== Classifier model (full training set) ===

RandomForest

Bagging with 100 iterations and base learner

weka.classifiers.trees.RandomTree -K 0 -M 1.0 -V 0.001 -S 1 -do-not-check-capabilities

Time taken to build model: 1.58 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         553               14.1795 %
Incorrectly Classified Instances      3347               85.8205 %
Kappa statistic                         -0.0016
Mean absolute error                      0.2452
Root mean squared error                  0.358
Relative absolute error                100.1316 %
Root relative squared error            102.324  %
Total Number of Instances             3900
```

Figure 12: RandomForest full training set

Based on the data presented in Figure 11 we can see that out of a total of 3900 instances only 553 were classified correctly which accounts for a 14.1795% accuracy. In contrast a significant majority of 3347 instances or 85.8205% were classified incorrectly. The Kappa Statistic, which measures the agreement, between predicted and actual classifications is dangerously low at 0.0016 indicating a lack of agreement and falling far from expectations. The Mean Absolute Error stands at 0.2452 representing the magnitude of errors in the predictions regardless of their direction. Additionally the Root Mean Squared Error measures differences between predicted and observed values. Is recorded at 0.358. Moreover both the Relative Absolute Error (100.1316%) and Root Relative Squared Error (102.324%) are alarmingly high as they represent normalized measures of error between predicted and actual values. Taking these metrics into consideration makes it evident that the models performance does not meet expectations suggesting adjustments may be necessary, in either the model or its parameters to achieve results.

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.138	0.129	0.148	0.138	0.143	0.010	0.500	0.147	Fortnightly
	0.113	0.130	0.122	0.113	0.118	-0.018	0.490	0.136	Weekly
	0.128	0.150	0.128	0.128	0.128	-0.022	0.482	0.140	Annually
	0.155	0.147	0.150	0.155	0.152	0.007	0.500	0.148	Quarterly
	0.165	0.136	0.165	0.165	0.165	0.028	0.505	0.149	Bi-Weekly
	0.132	0.139	0.136	0.132	0.134	-0.007	0.497	0.140	Monthly
	0.161	0.170	0.143	0.161	0.151	-0.009	0.474	0.143	Every 3 Months
Weighted Avg.	0.142	0.143	0.142	0.142	0.142	-0.002	0.492	0.143	

Figure 13: Random forest detailed accuracy by class

The table above (figure 13), provides a detailed breakdown of accuracy metrics for a classifier, organized by various classes representing different frequencies of events or actions, including "Fortnightly," "Weekly," "Annually", "Quarterly", "Hi-Weekly" and "Monthly". These metrics provide insights into the classifier's performance for each class. The True Positive Rate (TP Rate), also known as recall, assesses how effectively the model identifies actual positives. For instance, the "Fortnightly" class achieved a TP rate of 0.138, indicating that 13.8% of actual fortnightly cases were correctly identified. Conversely, the False Positive Rate (FP Rate) gauges the model's tendency to misclassify actual negatives as positives. In the case of the "Weekly" class, it showed an FP rate of 0.130, implying that 13% of non-weekly cases were incorrectly categorized as weekly. Precision, another critical metric, reveals the proportion of positive identifications that were genuinely correct.

Specifically, the "Annually" class, with a TP rate precision of 0.128, suggests that 12.8% of the instances classified as annually were accurate. These metrics, along with others like F-Measure, Matthews Correlation Coefficient (MCC), ROC Area, and PRC Area, provide a comprehensive evaluation of the classifier's performance across various classes. Overall, the classifier's performance appears suboptimal for all classes, consistently displaying low values in precision, recall, F-measure, and MCC, indicating a performance level not significantly better than random guessing.

```
=== Confusion Matrix ===

  a   b   c   d   e   f   g   <-- classified as
 75  73  68  71  84  81  90 |   a = Fortnightly
 80  61  83  84  70  78  83 |   b = Weekly
 57  76  73  94  85  86 101 |   c = Annually
 69  78 102  87  77  61  89 |   d = Quarterly
 77  68  68  81  90  63 100 |   e = Bi-Weekly
 67  74  89  78  70  73 102 |   f = Monthly
 83  69  88  84  71  95  94 |   g = Every 3 Months
```

Figure 14: Random Forest confusion matrix

This dataset details a confusion matrix, commonly employed in machine learning to assess the efficacy of classification algorithms. It visualizes the accuracy of predictions, with each matrix row representing actual class instances and each column indicating predicted class instances. The dataset includes seven classes, namely Fortnightly, Weekly, Annually, Quarterly, Bi-Weekly, Monthly, and Every 3 Months, which could pertain to various time-related events or measurements. The matrix values reflect the count of predictions by the model. For instance, the 'Fortnightly' row shows 75 accurate predictions but also 73 instances where 'Fortnightly' was incorrectly predicted as 'Weekly', and similar misclassifications in other categories. Correct predictions are denoted by diagonal values (like 75 for Fortnightly), while errors are shown in off-diagonal values. A high diagonal number signifies effective performance for that class due to more accurate predictions, whereas large off-diagonal numbers highlight confusion areas where the model frequently misclassifies one class as another, such as confusing 'Annually' with 'Every 3 Months'. This matrix is instrumental in discerning not only the overall performance of the model but also in identifying specific classes that are commonly misclassified, suggesting the need for model refinement or additional training data for these problematic categories. The data indicates that while the model generally distinguishes between classes effectively, it still struggles with certain categories, highlighting areas for improvement.

Decision Table

```
=== Run information ===

Scheme:       weka.classifiers.rules.DecisionTable -X 1 -S "weka.attributeSelection.BestFirst -D 1 -N 5"
Relation:     shopping_trends_updated
Instances:    3900
Attributes:   18
              Customer ID
              Age
              Gender
              Item Purchased
              Category
              Purchase Amount (USD)
              Location
              Size
              Color
              Season
              Review Rating
              Subscription Status
              Shipping Type
              Discount Applied
              Promo Code Used
              Previous Purchases
              Payment Method
              Frequency of Purchases
Test mode:    5-fold cross-validation
```

Figure 15: Decision table run information

Based on the illustration above (figure 15). Each record is detailed with 18 distinct attributes and 3,900 instances that provide a holistic view of customer shopping behavior. These attributes range from the Customer ID, serving as a unique identifier, to various demographic details like age and gender. It also includes specifics of the shopping experience, such as the item purchased, its category, the purchase amount in USD, and customer-specific information like location, size, and color of the item. Additional details are covered, including the season of purchase, review rating, subscription status, shipping type, application of discounts, use of promo codes, the customer's previous purchase history, preferred payment method, and their frequency of purchases.

To ensure the robustness of the dataset in modeling, a 5-fold cross-validation technique is employed. This approach involves partitioning the dataset into five segments, using each in turn for testing while the remaining segments are used for training. Overall, this run information is evidently designed for in-depth analysis of customer shopping behaviors and preferences, offering valuable insights for areas such as marketing, sales forecasting, and customer relationship management.

```
=== Classifier model (full training set) ===

Decision Table:

Number of training instances: 3900
Number of Rules : 25
Non matches covered by Majority class.
        Best first.
        Start set: no attributes
        Search direction: forward
        Stale search after 5 node expansions
        Total number of subsets evaluated: 93
        Merit of best subset found:   16.564
Evaluation (for feature selection): CV (leave one out)
Feature set: 9,18

Time taken to build model: 0.53 seconds
```

Figure 16: Decision table classifier model full training set

The core of the model is a Decision Table, a method for representing and analyzing rules applied to data. The model's training involved 3,900 instances, serving as the data points for training. From these, it derived 25 rules that are pivotal for making predictions or decisions on new data. In instances where none of these 25 rules apply, decisions are based on the majority class in the dataset, a typical approach in decision tables to manage unmatched data points.

The model's attribute selection process involved a "best first" forward search algorithm. It began with no attributes and incrementally added them, assessing the performance at each step, to construct the most effective subset for predictions. The search was halted if no improvement was observed after five node expansions in the search tree. In total, 93 attribute subsets were evaluated, with the best subset achieving a merit score of 16.564, reflecting its predictive efficiency.

Furthermore, the model employed leave-one-out cross-validation (CV) for feature selection, a meticulous approach where each dataset instance is used once as the validation set, while the rest form the training set. This cycle repeats for every instance in the dataset. The model identified attributes 9 and 18 as most effective for predictions, corresponding to specific data fields in the dataset. Finally, the model's construction required 0.53 seconds, indicating the computational time taken using the given training data. This summary underscores the model's focus on efficiently selecting relevant features from the dataset to enhance accuracy in predictions or decisions, utilizing the decision table methodology.

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         541               13.8718 %
Incorrectly Classified Instances      3359               86.1282 %
Kappa statistic                         -0.0056
Mean absolute error                      0.245
Root mean squared error                  0.351
Relative absolute error                100.0427 %
Root relative squared error            100.3085 %
Total Number of Instances             3900
```

Figure 17: Decision table stratified cross validation

The dataset summary provided an insight into the performance of a machine learning model evaluated using stratified cross-validation. This method involves dividing the dataset into various parts or folds, with each fold representing the overall dataset's stratification. The model undergoes training on several of these folds and is then tested on the remaining ones. This process is repeated to thoroughly assess the model's performance.

In this case, the model correctly classified 541 out of the 3,900 instances, equating to roughly 13.87% of the dataset, indicating the model's accuracy level. However, a significant portion, about 86.13% or 3,359 instances, were incorrectly classified, pointing to a high error rate in the model's predictions. The Kappa statistic for the model is -0.0056, suggesting minimal or no agreement between the predicted and actual classifications, beyond what might occur by chance.

Additionally, the Mean Absolute Error is noted as 0.245, representing the average of absolute differences between predicted and actual values, where a lower figure would indicate better performance. The Root Mean Squared Error stands at 0.351, another measure of accuracy, implying the average squared difference between the model's predictions and actual observations. The Relative Absolute Error and Root Relative Squared Error are 100.0427% and 100.3085%, respectively. These metrics compare the model's errors to those of a basic model that always predicts the average outcome. Values around or exceeding 100% suggest that the model performs no better, or even worse, than this simple model. The total number of instances in the dataset is 3,900.

In summary, the metrics collectively indicate that the model is underperforming, with a low accuracy rate and error measurements suggesting its performance is comparable to or poorer than a basic naive model.

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.118	0.121	0.136	0.118	0.127	-0.003	0.493	0.137	Fortnightly
	0.135	0.139	0.135	0.135	0.135	-0.004	0.500	0.138	Weekly
	0.224	0.238	0.139	0.224	0.171	-0.012	0.492	0.142	Annually
	0.256	0.253	0.146	0.256	0.186	0.003	0.488	0.144	Quarterly
	0.051	0.062	0.118	0.051	0.071	-0.016	0.513	0.139	Bi-Weekly
	0.074	0.087	0.124	0.074	0.093	-0.016	0.489	0.140	Monthly
	0.108	0.106	0.152	0.108	0.126	0.002	0.486	0.148	Every 3 Months
Weighted Avg.	0.139	0.144	0.136	0.139	0.130	-0.006	0.494	0.141	

Figure 18: Decision table detailed accuracy by class

The table (figure 18) provides a comprehensive analysis of the accuracy of a classification model, broken down by individual classes, such as 'Fortnightly', 'Weekly', 'Annually', and others. Each row in this dataset represents a distinct class, with columns detailing various metrics that assess the model's performance for each specific class. The True Positive Rate (TP Rate) indicates the percentage at which the model accurately identifies a class; for example, a TP rate of 0.118 for 'Fortnightly' implies that 11.8% of cases in this class were correctly identified. The False Positive Rate (FP Rate) shows the frequency of incorrect class identifications by the model, as observed in the 12.1% FP rate for 'Fortnightly'. Precision, another metric, represents the proportion of true positives in all instances classified under a particular class, like the 13.6% precision for 'Fortnightly'. Recall, synonymous with TP Rate or sensitivity, measures the model's ability to find all relevant cases in a class.

The F-Measure combines precision and recall into a singular metric, providing a balanced view of the model's performance. The Matthews Correlation Coefficient (MCC) is a robust measure of the quality of binary classifications, adaptable to multiclass classifications, and considers both positive and negative predictions. The ROC Area metric, reflecting the probability that a positive instance is ranked higher than a negative one, shows values close to 0.5 in this dataset, indicating a lack of discriminative power. The Precision-Recall Curve Area (PRC Area) is particularly useful in scenarios of class imbalance. Each row in the dataset corresponds to a 'Class', the specific category or label the model aims to predict. Additionally, the "Weighted Avg." row presents a weighted average of each metric across all classes, factoring in the number of instances in each class.

Observationally, the model exhibits generally low performance across all classes, as seen in the low TP rates, precision, and F-measure values. The ROC Area values hovering around 0.5 suggest performance equivalent to random guessing. Moreover, negative MCC values for most classes indicate a general disagreement between the model's predictions and the actual classifications. Overall, this data points to the model's suboptimal performance in classifying the given classes, signaling a need for re-evaluation or improvement in its approach.

```
=== Confusion Matrix ===

  a   b   c   d   e   f   g   <-- classified as
 64  78 132 147  28  41  52 |   a = Fortnightly
 62  73 120 149  31  54  50 |   b = Weekly
 73  76 128 142  38  50  65 |   c = Annually
 65  80 136 144  29  48  61 |   d = Quarterly
 77  73 135 132  28  41  61 |   e = Bi-Weekly
 57  76 135 139  42  41  63 |   f = Monthly
 71  84 135 134  41  56  63 |   g = Every 3 Months
```

Figure 19: Decision table confusion matrix

The table above (figure 19) illustrates a confusion matrix. This matrix is structured with rows and columns, where each row represents instances of an actual class and each column denotes instances as classified by the model. The specific classes in this dataset include 'Fortnightly', 'Weekly', 'Annually', 'Quarterly', 'Bi-Weekly', 'Monthly', and 'Every 3 Months'. In this matrix, rows correspond to the actual class, with the first row, for example, showing instances genuinely classified as 'Fortnightly'. Columns reflect the model's predictions, such as the first column indicating instances predicted as 'Fortnightly'. The diagonal values, stretching from the top left to the bottom right, signify correct classifications by the model. For instance, a '64' in the first cell indicates 64 'Fortnightly' instances correctly classified. Conversely, off-diagonal values represent misclassifications, like '78' in the second column of the first row, showing 'Fortnightly' instances incorrectly classified as 'Weekly'. A detailed analysis reveals substantial misclassifications across various classes, particularly between 'Fortnightly', 'Quarterly', and 'Annually', and similar confusions among other classes. This pattern suggests the model's difficulty in effectively distinguishing between these categories. Overall, the confusion matrix highlights the model's struggle with accurate classification, especially in differentiating certain classes.

K-Star

```
=== Run information ===

Scheme:       weka.classifiers.lazy.KStar -B 20 -M a
Relation:     shopping_trends_updated
Instances:    3900
Attributes:   18
              Customer ID
              Age
              Gender
              Item Purchased
              Category
              Purchase Amount (USD)
              Location
              Size
              Color
              Season
              Review Rating
              Subscription Status
              Shipping Type
              Discount Applied
              Promo Code Used
              Previous Purchases
              Payment Method
              Frequency of Purchases
Test mode:    5-fold cross-validation
```

Figure 20: K-Star model run information

The dataset in figure 20 consists of 3,900 instances, each representing individual customer shopping experiences, and is analyzed using the KStar. The KStar algorithm is a type of lazy learning, specifically an instance-based learning algorithm, which means it doesn't build a general internal model but makes predictions based on the specific instances of the data it encounters. This approach is particularly suited for datasets with complex, non-linear relationships between attributes.

In this dataset, each record is detailed with 18 attributes, including the customer's ID, age, gender, details of the item purchased (such as category, size, color), purchase amount, the location of the purchase, the season in which the purchase was made, the customer's review rating, subscription status, type of shipping, discounts and promo codes used, previous purchase history, payment method, and the frequency of purchases. These attributes provide a comprehensive view of each shopping experience and customer profile. Using the KStar algorithm, this dataset likely aims to uncover patterns and trends in customer behavior, preferences, and purchasing decisions. The use of 5-fold cross-validation in testing ensures a robust evaluation of the model's performance, reducing the likelihood of overfitting and improving the generalizability of the findings. The specified parameters in the algorithm (-B 20 -M a) might be tuning the behavior of the algorithm, like adjusting the basis of similarity between instances or the manner in which instances influence each other. This analysis could be valuable for understanding consumer behavior, enhancing customer experience, and informing business strategies.

```
=== Classifier model (full training set) ===

KStar Beta Verion (0.1b).
Copyright (c) 1995-97 by Len Trigg (trigg@cs.waikato.ac.nz).
Java port to Weka by Abdelaziz Mahoui (am14@cs.waikato.ac.nz).

KStar options : -B 20 -M a

Time taken to build model: 0 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances         560               14.359  %
Incorrectly Classified Instances      3340               85.641  %
Kappa statistic                          0.0007
Mean absolute error                      0.2442
Root mean squared error                  0.4228
Relative absolute error                 99.7408 %
Root relative squared error            120.8233 %
Total Number of Instances             3900
```

Figure 21: K-Star classifier model full training set

The dataset description above (figure 21), provided pertains to a training session of a machine learning model using the KStar classifier, the KStar classifier operates by comparing new instances to training instances to determine classification. The specific parameters mentioned, -B 20 and -M a, likely relate to blending settings and the handling of missing values, respectively. Remarkably, the model's construction time is reported as 0 seconds, suggesting either an extremely small dataset or highly efficient processing.

The performance metrics of the model reveal several critical insights. The model correctly classified only 14.359% of instances, while incorrectly classifying 85.641%, indicating a significant limitation in its predictive accuracy. The kappa statistic value is near zero, which implies that there is almost no agreement between the predicted and actual classifications. Other metrics like the mean absolute error and root mean squared error provide measures of the average errors in the model’s predictions. The relative absolute error and root relative squared error are normalized error metrics, further emphasizing the model's lack of accuracy. Finally, the total number of instances in the dataset is 3900, which the model used for training and testing.

In summary, the dataset description suggests that the model's performance is notably poor. The high rate of incorrect classifications and the low kappa statistic highlight that the model's predictive ability is barely above random chance, indicating a need for further refinement or a different modeling approach to improve its classification accuracy.

=== Detailed Accuracy By Class ===

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.149	0.133	0.153	0.149	0.151	0.016	0.507	0.148	Fortnightly
	0.121	0.132	0.128	0.121	0.124	-0.012	0.505	0.141	Weekly
	0.142	0.149	0.140	0.142	0.141	-0.008	0.504	0.153	Annually
	0.153	0.142	0.154	0.153	0.153	0.011	0.514	0.151	Quarterly
	0.133	0.136	0.138	0.133	0.136	-0.003	0.497	0.138	Bi-Weekly
	0.190	0.149	0.174	0.190	0.182	0.039	0.530	0.160	Monthly
	0.118	0.157	0.117	0.118	0.118	-0.039	0.483	0.141	Every 3 Months
Weighted Avg.	0.144	0.143	0.143	0.144	0.143	0.000	0.506	0.147	

Figure 22: K-Star detailed accuracy by class

The dataset accuracy in figure 22 offers an extensive analysis of a classification model's accuracy across various frequency-related categories (such as Fortnightly, Weekly, Annually, etc.). It utilizes several key metrics to evaluate the model's performance for each category. The True Positive Rate (TP Rate) reflects the percentage of accurately identified positive instances per category. For instance, a TP rate of 14.9% for the Fortnightly category indicates that 14.9% of its instances were correctly classified. The False Positive Rate (FP Rate) indicates the proportion of negative instances erroneously classified as positive, such as 13.3% of non-Fortnightly instances mislabeled as Fortnightly.

Precision measures the accuracy of positive predictions, demonstrated by the fact that 15.3% of instances labeled as Fortnightly were correct. Recall, identical to TP Rate, shows the fraction of true positive instances out of all actual positives, also at 14.9% for Fortnightly. The F-Measure, a harmonic mean of precision and recall, assesses the balance between these two metrics. The Matthews Correlation Coefficient (MCC) offers a quality measure for binary classifications, considering both true and false positives and negatives.

Two additional metrics, the ROC Area and the PRC Area, evaluate the model's discriminative ability and the trade-off between precision and recall respectively. However, the overall assessment indicates substandard performance, with low scores in TP Rate, Precision, Recall, F-Measure, and MCC, and ROC Area values around 0.5. This suggests the model's ineffectiveness in distinguishing between classes, operating at a level comparable to random chance.

```
=== Confusion Matrix ===

  a   b   c   d   e   f   g   <-- classified as
 81  73  71  66  81  85  85 |   a = Fortnightly
 74  65  79  82  69  90  80 |   b = Weekly
 87  76  81  89  78  82  79 |   c = Annually
 61  91 101  86  65  70  89 |   d = Quarterly
 86  64  74  72  73  81  97 |   e = Bi-Weekly
 61  71  81  77  67 105  91 |   f = Monthly
 79  69  91  88  97  91  69 |   g = Every 3 Months
```

Figure 23: K-Star confusion matrix

Based on this matrix (figure 23), the classes (a, b, c, d, e, f, g) represent different frequencies like Fortnightly, Weekly, Annually, and so on. Each matrix cell shows the count of instances for a specific classification. For example, a cell count of 81 in the 'Fortnightly' row indicates that 81 Fortnightly instances were correctly predicted as such. Conversely, off-diagonal cells indicate misclassifications, such as 73 Fortnightly instances being wrongly classified as Weekly.

Each row in the matrix provides insights into a specific class:

1. 'Fortnightly': 81 correct predictions, with remaining instances misclassified.
2. 'Weekly': 65 instances correctly identified.
3. 'Annually': 81 correct classifications.
4. 'Quarterly': 86 instances accurately predicted.
5. 'Bi-Weekly': 73 correctly classified.
6. 'Monthly': This class has the highest accuracy with 105 correct predictions.
7. 'Every 3 Months': 69 instances correctly classified.

Key takeaways from this matrix include:

- The diagonal elements (81, 65, 81, 86, 73, 105, 69) show the number of correct classifications per class.
- High numbers in off-diagonal cells highlight significant misclassification, suggesting areas for model improvement.
- The model is most effective in predicting 'Monthly' instances, evidenced by the highest diagonal value in its row.

Overall, this confusion matrix is essential for pinpointing the model's strengths and weaknesses, helping to refine training and feature selection for better accuracy.

Reference:

https://www.kaggle.com/datasets/iamsouravbanerjee/customer-shopping-trends-dataset