**3. Methodology**

**3.1. Statistical Methods**

The statistical methods in this study were vital for preparing the data and understanding customer behavior. The process began with cleansing the data by removing duplicates, handling missing values, and addressing outliers to ensure the dataset was accurate and reliable. New features were created, including time-based attributes and customer behavior metrics like Recency, Frequency, and Monetary (RFM) values, to better understand customer interactions. Correlation analysis was then applied to explore relationships between features and their impact on customer value. These steps provided a clean and enriched dataset, forming a strong foundation for advanced modeling and analysis.

**3.1.1. Data Loading and Exploration**

The evaluation started by gathering information from two Excel sheets containing online retail transaction records from 2009 to 2011. These datasets were combined into one to allow for a complete analysis of all transactions over the three years.

The first step was to explore the structure of the data. The initial rows were reviewed to check how the information was organized and to spot potential problems. The column names, data types, and the total number of rows and columns, were inspected to understand the dataset better. Basic calculations, like averages, minimums, maximums, and standard deviations, were done to identify trends, patterns, or irregularities in the numbers.

To make the data accurate and ready for analysis, several issues were addressed:

* **Missing Values**: Missing entries were removed, especially in critical columns like CustomerID. This ensured that customer-level analysis remained reliable and meaningful.
* **Duplicates**: Duplicate rows were identified and removed to avoid inaccurate results and to maintain a clean dataset.
* **Irregularities**: Errors, such as negative values in Quantity or Price, were identified and removed. These were likely mistakes in data entry that could affect the integrity of the analysis.
* **Unnecessary Columns**: Columns that were not relevant to the goals of the evaluation were dropped. This helped focus the dataset on useful information and reduced complexity.

**Key Observations**

* Many records lacked “CustomerID”, which was crucial for customer-specific evaluations. These were handled appropriately to maintain the reliability of the results.
* Negative values in “Quantity” and “Price” were found, indicating potential errors. These were eliminated to ensure accuracy.
* Irrelevant columns were removed to make the dataset more manageable and focused.

After addressing these issues, the dataset was cleaned, simplified, and ready for the next steps of feature creation and model development.

**3.1.2. Data Cleaning and Feature Engineering**

Clean the data to make it ready for analysis. Duplicate rows were removed, and records missing critical information, such as “CustomerID”, were excluded, as these entries could not be linked to specific customers. Entries with negative values in the Quantity or Price columns were also eliminated, as these likely represented errors or returns, which could distort the results.

Once the data was cleaned, new features were created to enhance its usefulness:

* **Total\_Amount**: Calculated by multiplying Price and Quantity, this feature represented the total value of each transaction.
* **Hour, Day\_of\_Week, and Month**: These features were extracted from transaction dates to highlight patterns in customer shopping behavior.
* **Is\_Weekend**: This feature indicated whether the purchase occurred on a weekend, making it easier to compare weekday and weekend shopping trends.

To better understand customer behavior, **Recency, Frequency, and Monetary (RFM) Analysis** was conducted:

* **Recency**: how recently a customer made a purchase.
* **Frequency**: how often they bought something.
* **Monetary Value**: how much they spent.

Additional metrics, such as the average purchase value, the smallest and largest amounts spent, and variations in spending, were calculated for each customer to provide a more detailed view of individual buying habits.

The dataset was further analyzed to assess how often customers made purchases. For each customer, the average time between purchases and the variability in these intervals were calculated. This analysis revealed whether customers shopped consistently or sporadically.

Finally, outliers were addressed. Extreme values, such as unusually high or low transaction amounts, were capped to a reasonable range to prevent them from skewing the analysis or affecting the predictive models.

By the end of these steps, the dataset was fully cleaned and enriched with meaningful features. This prepared the data for the next stages of analysis, providing a solid foundation for building reliable and accurate predictive models.

**3.1.3 Correlation Analysis**

I conducted a correlation analysis to uncover relationships between the key features in the dataset. This step helped me understand how certain variables influenced customer value and behavior. For instance, I looked at whether higher purchase quantities were associated with higher spending or if specific time-based features like weekends impacted customer activity.

This analysis highlighted important patterns. Features like Recency, Frequency, and Monetary Value showed strong relationships with customer spending habits, helping to identify which factors played a significant role in determining customer value. The findings from this step provided valuable insights that shaped the development of predictive models.

**3.2 Machine Learning Methods**

The dataset contains a variety of data types, including categorical variables, and has huge amounts of missing values. XGBoost was chosen as the machine learning algorithm for this project because it effectively handles these challenges. It can process structured data with mixed types and automatically manage missing values during training, eliminating the need for extensive preprocessing.

Correlation analysis is useful for understanding linear relationships between features, but it has its limitations when it comes to capturing complex interactions. XGBoost excels in this area by identifying both linear and non-linear relationships, making it a more powerful tool for predictive tasks. With its ability to model intricate feature interactions, XGBoost provided a robust solution for analyzing this dataset and predicting customer value accurately.

### **3.2.1. XGBoost Model Development**

The XGBoost algorithm was chosen for its exceptional ability to handle structured data efficiently and model complex feature interactions. It stands out for its gradient boosting framework, which iteratively improves model accuracy by minimizing prediction errors. XGBoost’s inherent flexibility and scalability make it an ideal choice for datasets with diverse data types and intricate relationships among features.

To maximize its performance, hyperparameter tuning was performed using the Optuna framework, a powerful optimization tool designed to find the best model configurations. This process systematically tested different combinations of parameters to identify the optimal settings that would enhance model accuracy and generalization. The key parameters tuned included:

* **max\_depth**: Controlled the maximum depth of the decision trees. A deeper tree allows the model to learn more detailed patterns but increases the risk of overfitting. Optuna determined the ideal depth to balance underfitting and overfitting.
* **learning\_rate**: Adjusted the step size for gradient updates. A smaller learning rate improves accuracy by taking smaller steps during optimization, while a larger rate speeds up training. Optuna optimized this parameter to achieve the perfect balance between training speed and precision.

Through this tuning process, the XGBoost model was fine-tuned to capture essential patterns in the data. It effectively learned both simple and complex relationships among features, enabling it to provide accurate predictions while maintaining robust generalization to unseen data. This optimization process ensured the model was reliable, efficient, and well-suited to the specific requirements of predicting customer value.

**3.2.2. Feature Importance Analysis**

After training the XGBoost model, feature importance analysis was conducted to uncover the key variables driving the predictions. This analysis revealed which features had the greatest impact on customer value, providing actionable insights into customer behavior and purchasing patterns.

The importance of each feature was measured based on how often it was used in the model’s decision trees and the magnitude of its contribution to reducing prediction errors. Features with higher importance scores played a significant role in shaping the model’s outcomes, making them valuable for further exploration.

Key findings highlighted the critical role of certain variables:

* Transaction Behaviors: Metrics like Total\_Amount, Recency, and Frequency stood out as major contributors, emphasizing the importance of understanding customer spending patterns and engagement levels.
* Time-Based Features: Variables such as Day\_of\_Week, Month, and Is\_Weekend provided insights into when customers were most active, helping to identify seasonal or weekly shopping trends.

This step not only strengthened the predictive power of the model but also provided practical insights for decision-making. By identifying the most influential features, the analysis offered a deeper understanding of the factors driving customer value, which could be used to inform business strategies and enhance customer targeting.

**3.3 Deep Learning Methods**

A Neural Network was chosen for its ability to model complex, non-linear relationships in the data, making it ideal for capturing intricate patterns in customer behaviors and interactions. Unlike simpler methods like logistic regression or decision trees, Neural Networks automatically learn feature interactions during training, reducing the need for manual feature engineering.

This model also complements XGBoost, providing an alternative approach to ensure no significant patterns are overlooked. While methods like Random Forest or SVMs can work well, they are often less effective at handling non-linear relationships or scaling with larger datasets. By including a Neural Network, the project benefits from a versatile and powerful tool capable of uncovering deeper insights into customer value.

**3.3.1. Neural Network Development**

A deep neural network was developed to complement the XGBoost model, leveraging its ability to capture complex, non-linear relationships within the data. This approach provided a robust framework for analyzing intricate patterns and interactions among features, ensuring no valuable insights were overlooked.

The architecture of the neural network consisted of:

* **Three Dense Layers**: These layers enabled the model to process and learn from multiple levels of complexity in the data. The dense connections allowed the network to identify both simple and deep patterns, making it highly effective for predicting customer value.
* **Non-linear Activation Functions**: Each layer used non-linear activation functions, which helped the network model relationships beyond straightforward linear patterns, enhancing its predictive power.

To prevent overfitting and improve the model’s generalization to unseen data, **early stopping** was implemented. During training, the network’s performance on the validation set was continuously monitored. If the performance stopped improving, the training process was automatically halted. This approach ensured that the model did not overfit to the training data and maintained its ability to make accurate predictions on new data.

The combination of this carefully designed architecture and overfitting prevention techniques resulted in a neural network that was not only powerful but also reliable, making it a valuable component of the predictive modeling process.

**3.3.2. Learning Curve Analysis**

To evaluate the performance of the Neural Network, learning curves were generated, showing how the model’s accuracy and error changed during training and validation over successive epochs. These plots provided crucial insights into the model's behavior and helped identify potential issues such as overfitting or underfitting.

**Overfitting** occurs when the model performs exceptionally well on the training data but struggles with validation data, indicating that it has memorized patterns specific to the training set rather than learning general trends. On the other hand, **underfitting** happens when the model fails to capture the patterns in both the training and validation data, resulting in poor performance.

The learning curves helped visualize these scenarios by comparing:

* **Training Performance**: Displayed how well the model was learning from the training data over time.
* **Validation Performance**: Showed how well the model generalized to unseen data during training.

Through these curves, adjustments were made to improve model stability, such as fine-tuning the network architecture, optimizing hyperparameters, or implementing regularization techniques. The goal was to ensure that both training and validation performance improved steadily without significant gaps, indicating a balanced model capable of generalizing effectively.

Learning curve analysis not only improved the reliability of the Neural Network but also provided a clear diagnostic tool for refining the training process and ensuring optimal performance.

**3.4. Model Evaluation and Metrics**

The evaluation focused on assessing the performance of both the XGBoost and Neural Network models using a set of consistent metrics. This process ensured a thorough comparison of their accuracy, reliability, and ability to predict customer value effectively.

Metrics Used:

* MSE measured the average of the squared differences between predicted and actual values. A smaller MSE value indicated higher accuracy in predictions, as it penalized larger errors more heavily.
* RMSE, derived from MSE, provided the average prediction error in the same units as the target variable. It offered an intuitive sense of how far predictions deviated from actual values.
* MAE calculated the average absolute differences between predicted and actual values. It was a straightforward metric to understand and highlighted the overall consistency of the model's predictions.
* The R² score measured how well the models explained the variance in the data. A score closer to 1 indicated a strong correlation between predictions and actual values, showcasing the model's ability to capture patterns in the dataset.

Feature Importance (XGBoost):

A feature importance plot was created to highlight the variables that had the most significant influence on the XGBoost model's predictions. This provided actionable insights into which features contributed the most to customer value.

Learning Curves (Neural Network):

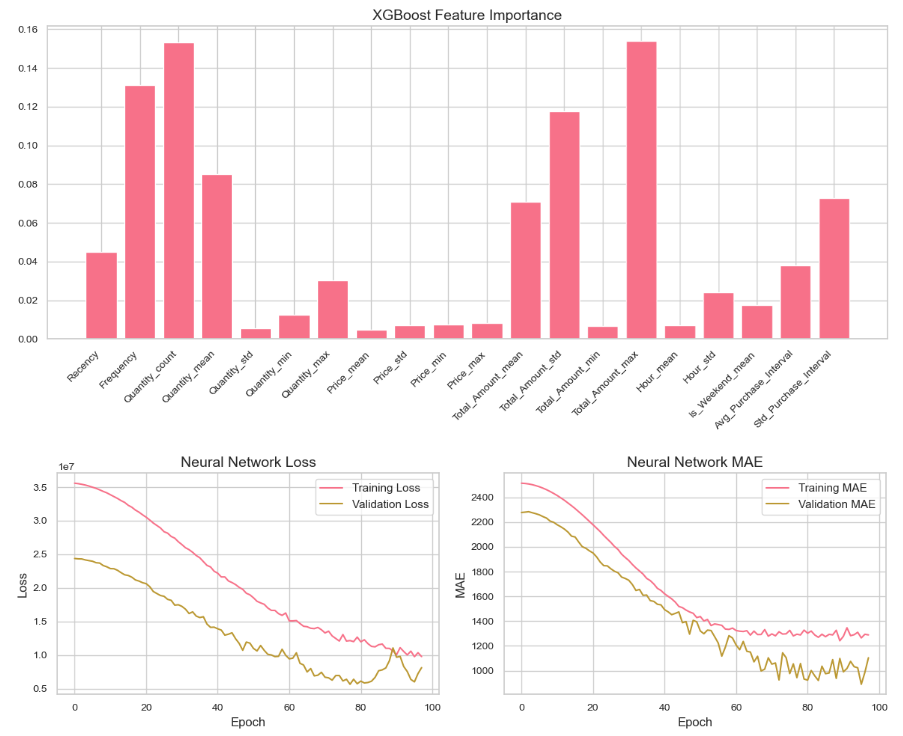
Learning curves were generated to track the Neural Network's performance during training and validation. These curves helped identify potential issues like overfitting (where the model performed well on training data but poorly on validation data) or underfitting (where the model failed to perform well on both).

Scatter Plots of Predicted vs. Actual Values:

Scatter plots compared the models' predicted values against the actual values. Points clustering around the diagonal line indicated high accuracy, while deviations highlighted areas needing improvement.

The combination of metrics and diagnostic tools provided a comprehensive evaluation of the models. MSE, RMSE, and MAE offered insights into the precision and consistency of predictions, while the R² score assessed how well the models captured the underlying patterns. Feature importance analysis shed light on influential variables, and visual diagnostics like learning curves and scatter plots illustrated the models' accuracy and learning progress.

These evaluations ensured that both the XGBoost and Neural Network models were rigorously tested, enabling the selection of the most effective approach for predicting customer value.



**3.5. Survey Data Processing and Integration**

Customer survey responses were included in the analysis through a structured approach to enhance the overall dataset. The survey data was first loaded from a CSV file, and inconsistencies in column names and formatting were resolved. This cleaning process ensured the data was well-organized and ready for further analysis.

Age ranges in the survey data were mapped to numerical values, converting qualitative categories into numeric data. This adjustment made the age information compatible with machine learning models and suitable for predictive analysis.

Feature engineering was applied to prepare the survey data effectively. Categorical variables, such as Gender, Country, and Shopping Frequency, were transformed using one-hot encoding. This process ensured the variables were machine-readable without losing their informational value. Numerical features, including Age, Membership Count, and Unified System Preference, were standardized to align their scales. This step ensured that no single variable disproportionately influenced the model predictions.

After processing, the survey data was integrated with the transaction data. This merged dataset combined customer transaction histories with survey insights, offering a comprehensive view of customer behavior and preferences. The unified dataset allowed for a richer and more accurate analysis.