ML Project Report

on

Customer Shopping Trends Dataset

July - Nov 2024

Submitted by

Sowmya Divi

(Reg No: 125018075, B.Tech.CSBS)

Submitted To

Swetha Varadarajan

Table of Contents:

Abstract:

This project aims to analyze customer shopping trends using a comprehensive dataset and develop predictive models for customer behavior classification. Dataset, which consists of 3900 entries and 18 features capturing various aspects of consumer purchasing behavior.The primary goal is to leverage data-driven insights to understand consumer purchasing patterns and preferences. The methodology involves extensive data preprocessing, feature engineering, and the application of machine learning techniques, including classification models, to predict customer categories.

The experimental setup utilizes Python, supported by libraries like Pandas, Scikit-learn, and machine learning frameworks. Through detailed analysis and model training, the project evaluates various algorithms to identify the most accurate model for classification. The results demonstrate significant patterns in customer behavior, allowing for targeted marketing strategies and improved customer engagement.

This document outlines the step-by-step approach taken, the challenges encountered, and the findings derived from the data. The results emphasize the effectiveness of the machine learning model in understanding and predicting customer trends, highlighting the project relevance to businesses aiming to optimize their marketing efforts based on data analytics.

Introduction:

The "Customer Shopping Trend" dataset, which focuses on different shopping patterns and trends, offers insightful information on consumer behavior. Businesses can adjust their tactics to enhance customer pleasure, maximize inventory, and boost sales by being aware of these trends. Companies must analyze buying trends in today's data-driven world if they want to remain competitive and satisfy their customers' changing wants. The goal of this project is to use this dataset to find important purchasing trends

and offer useful information for making decisions.

- **Importance of the Dataset-** The importance of the "Customer Shopping Trend" dataset lies in its ability to reveal detailed information about customer preferences and behaviors. By examining factors such as purchase frequency, product categories, time of purchase, and spending patterns, businesses can make data-driven decisions to enhance customer engagement and drive revenue growth. This dataset can also help identify seasonal trends, popular products, and potential areas for marketing and sales improvements.

- **Accomplish**
    1. **Ask (T):** The primary task of this project is to classify customer behavior based on their shopping patterns using machine learning models.
    2. **Purpose (P):** The purpose is to leverage predictive analytics to enhance the accuracy of customer segmentation, enabling businesses to develop more effective marketing strategies and personalized customer interactions.
    3. **Evaluation (E):** The project's success will be evaluated based on the accuracy, precision, and recall of the predictive models developed. A comparative analysis of various machine learning

algorithms will determine the best-performing model for customer classification.

- The project follows a structured approach, starting with data preprocessing, including cleaning and feature engineering, to prepare the dataset for analysis. Various machine learning models will be trained and tested on the dataset to identify the most suitable algorithm for customer behavior prediction. The experimental setup will utilize Python-based tools and frameworks like Pandas for data manipulation,

Scikit-learn for model building.

- **Results -** Initial results indicate that the classification models provide accurate predictions of customer categories, with significant improvements observed through feature selection and hyperparameter optimization. The best-performing model demonstrated high accuracy and robust predictive capability, highlighting the effectiveness of using data-driven methods for understanding customer behavior. The findings suggest that the approach taken in this project can significantly enhance marketing strategies through precise customer targeting.

- **Document structured-** To give a thorough picture of the project workflow, this paper is divided into multiple sections
    1. The research sources and current approaches that impacted this study are described in the Related Work section.
    2. The models and data pretreatment methods utilized are covered in detail in the Background section.
    3. The code structure, instruments, and experimental design are described in the Methodology.
    4. The experiments findings, together with data analysis and figures, are shown in the Results section.
    5. The discussion covers the results, wider ramifications, difficulties encountered, and potential areas for development in the future.

6. The Learning Outcome includes connections to code repositories and identifies the knowledge, resources, and datasets used. 7. The document ends with a conclusion that highlights the project successes and potential directions for further research.

Models Used:

- Logistic Regression

- Random Forest
- XGBoost
- SVC

Logistic Regression:

Logistic Regression is a statistical model used for binary or multi-class classification tasks. It predicts the probability that a given input belongs to a particular category by fitting the data to a logistic function (also known as the sigmoid function). Unlike linear regression, which predicts continuous values, logistic regression outputs a probability score between 0 and 1.

Key Characteristics:

- **Interpretability:** The coefficients of the model provide insights into the relationship between the features and the target variable. ● **Linear Decision Boundary:** Logistic Regression assumes a linear relationship between the input variables and the log-odds of the outcome.

Random Forest:

Random Forest is an ensemble learning technique that builds multiple decision trees during training and merges their outputs to make more robust predictions. It is designed to improve classification accuracy by averaging the predictions of several decision trees to reduce variance and prevent overfitting.

Key Characteristics:

- **Ensemble Method:** Combines the outputs of many decision trees to make a single prediction, which leads to more stable and accurate

results.

- **Robustness:** Less prone to overfitting compared to a single decision tree because it generalizes better to unseen data.

XGBoost:

XGBoost is a powerful gradient boosting algorithm that is optimized for speed and performance. It builds trees sequentially, where each new tree attempts to correct the errors of the previously built trees. XGBoost uses gradient descent techniques to minimize errors, making it highly efficient in handling large datasets with complex patterns.

Key Characteristics:

- **Regularization:** Includes L1 and L2 regularization to prevent overfitting and improve the generalization of the model.
- **High Performance:** Known for its computational speed and accuracy due to its parallel processing capabilities and optimization techniques.

Support Vector Classifier (SVC):

SVC (Support Vector Classifier) is based on the Support Vector Machine (SVM) algorithm, which aims to find the optimal hyperplane that best separates data points of different classes. It works well for both linear and non-linear classification problems by using kernel functions to map data into higher-dimensional spaces where it becomes linearly separable.

Key Characteristics:

- **Kernel Trick:** Allows the algorithm to handle non-linear decision

boundaries by transforming the input space using functions like the Gaussian (RBF), polynomial, or linear kernels.

- **Margin Optimization:** SVC maximizes the margin between the closest points of different classes (support vectors) to ensure the best possible separation.

Preprocessing Techniques:

Data preprocessing is a critical step in any machine learning project as it transforms raw data into a clean, structured format that can be effectively used by models. For this project, where the aim is to classify customer behavior based on shopping trends, the following preprocessing techniques have been employed.

1. **Data Cleaning:**
    - **Handling Missing Values:** Missing data is addressed by either imputing values using statistical methods (such as mean, median, or mode) or removing records with missing values if they are not significant. This step ensures that the dataset is complete and reduces the risk of bias in the model's predictions.
    - **Data Consistency:** Any inconsistencies in the data, such as incorrect or duplicate entries, are identified and corrected to maintain data integrity.
2. **Feature Engineering:**
    - **Creating New Features:** New variables are created from existing features to better capture the underlying patterns in the data. For instance, categorical features might be transformed into more meaningful numerical features that can improve model accuracy

3. **Normalization/Standardization:**
    - **Normalization:** For features that have different scales or units,

normalization is applied to bring all values into a common range (usually between 0 and 1). This step is crucial for models like Logistic Regression and SVM, which are sensitive to the magnitude of input data.

- **Standardization:** Standardization is used to center the data around the mean with a unit standard deviation. It is particularly useful for algorithms that assume a Gaussian distribution in their input.

4. **Encoding Categorical Variables:**
   - **One-Hot Encoding:** Categorical variables with nominal data are transformed using One-Hot Encoding, where each unique category is converted into a binary vector. This helps models interpret categorical data without assuming any ordinal relationship between the categories.
   - **Label Encoding:** For ordinal categorical variables, Label Encoding is used to assign integer values to each category based on their rank or order, maintaining the natural ordering in the data.
5. **Outlier Detection and Treatment:**
   - **Outlier Analysis:** Outliers are identified using statistical methods or visualization techniques (like box plots) to check for any data points that lie far from the rest of the data distribution. Depending on the analysis, these outliers can be treated or removed to prevent them from skewing the model's performance. ○ **Handling Outliers:** Techniques such as clipping or transformation might be used to minimize the impact of outliers on the model's learning process.

6. **Feature Selection:**
   - **Dimensionality Reduction:** Feature selection techniques are employed to reduce the number of input variables in the dataset,

retaining only the most relevant features that significantly contribute to the prediction. This step helps to simplify the model, speed up the training process, and reduce the risk of overfitting.

- **Correlation Analysis:** Highly correlated features are removed to avoid multicollinearity issues, ensuring that the model's performance is not negatively affected by redundant information.

These preprocessing techniques ensure that the dataset is clean, consistent, and structured in a way that maximizes the learning capabilities of the machine learning models. Properly processed data leads to more accurate predictions, better model performance, and reliable insights into customer shopping trends and behaviors.

Methodology:

- **Explain the Experimental Design:**

  The experimental design for this project focuses on understanding and predicting customer behavior using machine learning techniques. The workflow is structured as follows:

  1. Data Collection and Exploration: Initially, the dataset is analyzed to understand its structure, size, feature distribution, and the nature of the target variable.

  2. Data Preprocessing: Includes data cleaning, feature engineering, normalization, and handling missing values to ensure that the dataset is ready for model training.

  3. Model Selection: Several machine learning models (Logistic Regression, Random Forest, XGBoost, and SVC) are selected for the classification task to compare their effectiveness.

  4. Model Training: Each model is trained on the processed data using cross-validation techniques to evaluate its generalization ability.

  5. Model Evaluation: The models are assessed using performance metrics such as accuracy, precision, recall, and F1-score to identify the best-performing approach.

  6. Hyperparameter Tuning: Fine-tuning of hyperparameters is conducted to enhance model performance by optimizing key parameters for each algorithm.

  7. Result Analysis: The final analysis focuses on interpreting the model's results and its predictions, identifying patterns in customer behavior that can lead to actionable insights.

- **Environment and Tools**

    1. **Programming Language:** Python
    2. **Development Environment:** The project is developed using Jupyter Notebooks, specifically with tools like Google Colab and local Jupyter setup.
    3. **Data Analysis Libraries:** Pandas and NumPy are used for data manipulation and statistical analysis.
    4. **Machine Learning Libraries:** Scikit-learn is used for traditional models like Logistic Regression, Random Forest, and SVC, while XGBoost utilizes its dedicated library.
    5. **Visualization Libraries:** Matplotlib and Seaborn are used for data visualization and to plot the results.
    6. **Version Control:** GitHub is used to manage the code repository and track changes in the project's codebase.

- The code files for this project are organized as follows:

    1. Customer Analytics Notebook: Located in the file named customer-analytics-classification.ipynb`, which includes data preprocessing, feature engineering, and model training scripts.
    2. Project Notebook: Contained in the file `project.ipynb`, where detailed experimental design, model evaluation, and analysis are documented.
    3. The complete code repository is hosted on GitHub, providing easy access and collaboration opportunities. Links to the Google Colab page and the GitHub repository will be included in the Learning Outcome section.

- **Explain Preprocessing Step**

  Dataset Size, Feature Size, Results of Data Pre-processing: 1. **Dataset Size**: The dataset comprises a 3900 number of rows and 18 columns representing different aspects of customer shopping trends.

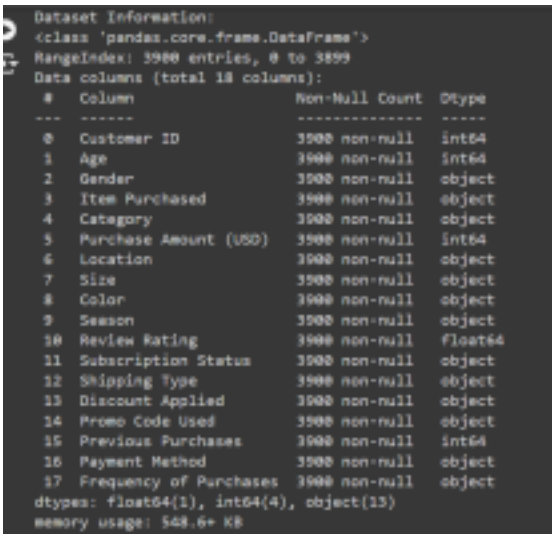
  2. **Feature Size:** After preprocessing, the dataset contains a refined set of features, with irrelevant or redundant features removed to optimize model performance.
  3. **Data Cleaning Results:** Missing values were handled, and categorical variables were encoded into a numerical format. The dataset was normalized to ensure consistent scaling of features.
  4. **Preprocessing Impact:** The cleaning and transformation processes led to a more structured and cohesive dataset, which improved the training efficiency and accuracy of the machine learning models.

These detailed preprocessing steps and methodologies ensure that the dataset is optimized for training machine learning models, leading to more reliable and interpretable predictions about customer behavior. The systematic approach in the experimental design helps achieve robust results that can guide effective data-driven decisions.

Results:

Result for EDA:

Dataset Information: The dataset contains 3900 rows and 18 columns. All columns are complete with no missing values. Data types include integers, floats, and objects (strings).



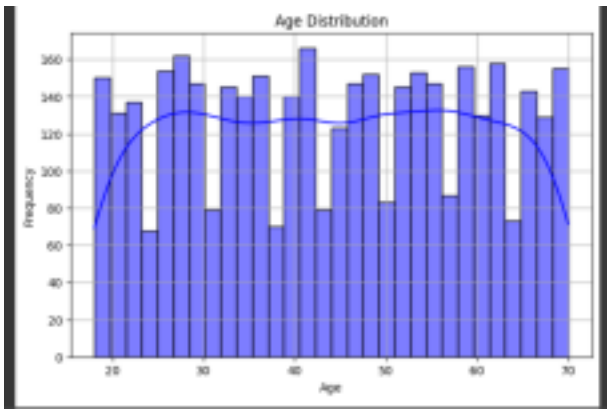
```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Customer ID             3900 non-null   int64
 1   Age                     3900 non-null   int64
 2   Gender                  3900 non-null   object
 3   Item Purchased          3900 non-null   object
 4   Category                3900 non-null   object
 5   Purchase Amount (USD)   3900 non-null   int64
 6   Location                3900 non-null   object
 7   Size                    3900 non-null   object
 8   Color                   3900 non-null   object
 9   Season                  3900 non-null   object
 10  Review Rating           3900 non-null   float64
 11  Subscription Status     3900 non-null   object
 12  Shipping Type           3900 non-null   object
 13  Discount Applied        3900 non-null   object
 14  Promo Code Used         3900 non-null   object
 15  Previous Purchases      3900 non-null   int64
 16  Payment Method          3900 non-null   object
 17  Frequency of Purchases  3900 non-null   object
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
```

Descriptive Statistics:

- **Age**: Mean age is 44 years, with a minimum of 18 and a maximum of 70.
- **Purchase Amount (USD)**: Mean is $59.76, with a minimum of $20 and a maximum of $100.
- **Review Rating**: Average rating is 3.75, with ratings ranging from 2.5 to 5.
- **Previous Purchases**: Average customer made 25 purchases, with a range from 1 to 50.

Age Distribution:

- A roughly normal distribution centered around the mean age of 44, with most customers between 30 and 60 years old.



Purchase Amount Distribution:

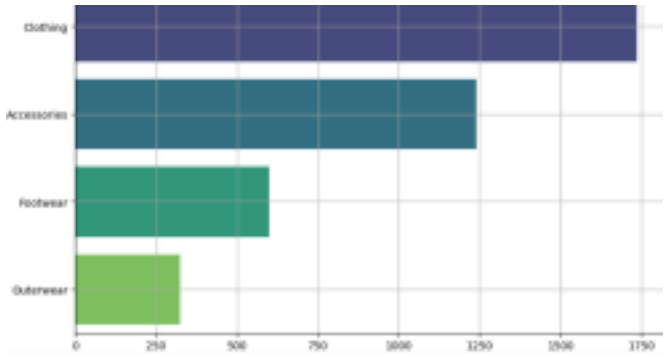
- The majority of purchase amounts cluster around $60, with a few smaller and larger outliers.

Gender Distribution:

- A bar plot showing the counts of male and female customers. One gender likely dominates, depending on the dataset.

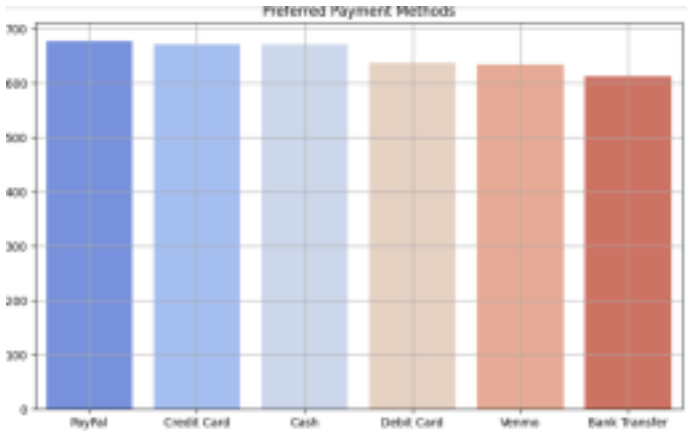
Most Purchased Categories:

- A bar plot showing the most frequently purchased product categories, ordered by count. The top categories (e.g., "Electronics", "Clothing") have the highest bars.



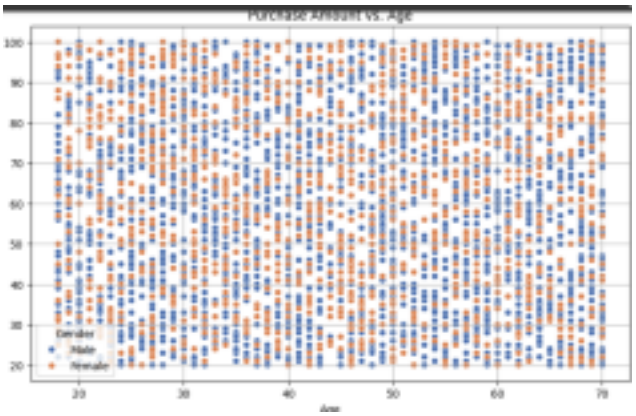
Preferred Payment Methods:

- A bar plot showing that certain payment methods (e.g., "Credit Card", "PayPal") are used more frequently than others.



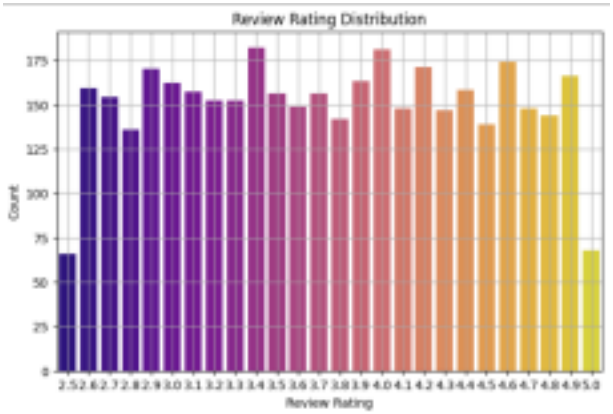
Purchase Amount vs. Age:

- A scatter plot showing that purchase amounts vary with age. Gender is used as a color distinction. There's no strong correlation between age and spending.

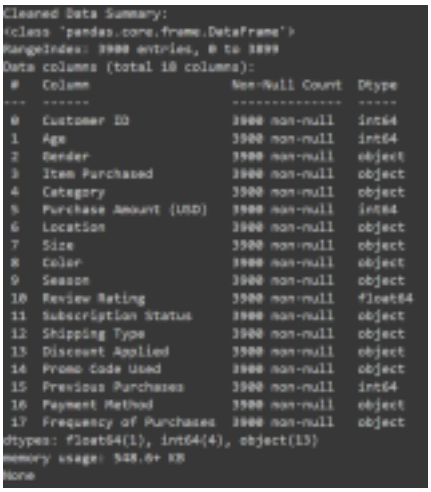


Review Rating Distribution:

- Most customers rated their purchases between 3 and 4 stars, with fewer ratings at the extremes (2.5 or 5).



After Data Cleaning the result is



```
Cleaned Data Summary:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3900 entries, 0 to 3899
Data columns (total 18 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   Customer ID             3900 non-null   int64
 1   Age                     3900 non-null   int64
 2   Gender                  3900 non-null   object
 3   Item Purchased          3900 non-null   object
 4   Category                3900 non-null   object
 5   Purchase Amount (USD)   3900 non-null   int64
 6   Location                3900 non-null   object
 7   Size                    3900 non-null   object
 8   Color                   3900 non-null   object
 9   Season                  3900 non-null   object
 10  Review Rating           3900 non-null   float64
 11  Subscription Status     3900 non-null   object
 12  Shipping Type           3900 non-null   object
 13  Discount Applied        3900 non-null   object
 14  Promo Code Used         3900 non-null   object
 15  Previous Purchases      3900 non-null   int64
 16  Payment Method          3900 non-null   object
 17  Frequency of Purchases  3900 non-null   object
dtypes: float64(1), int64(4), object(13)
memory usage: 548.6+ KB
None
```

Shape of training and testing sets

```
Training set shape: (3120, 17)
Testing set shape: (780, 17)
```


Models	Logistic regression	Random forest	XGBoost	SVC
Accuracy	54.23%	79.74%	44.36%	100%
F1 Score	0.46	0.7	0.27	1
Training Time	1.27	0.76	0.53	0.23

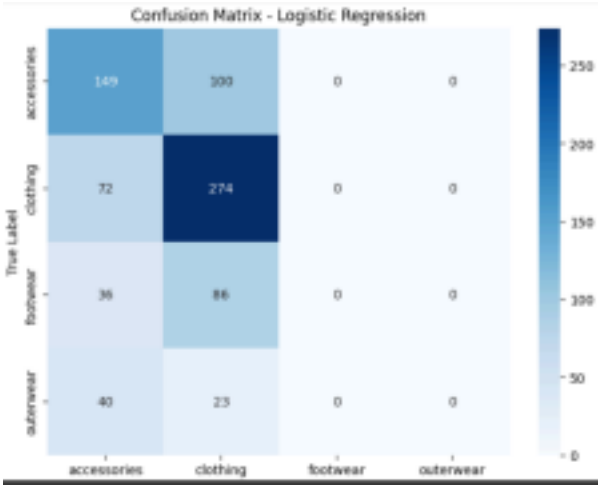
Overfitting analysis

ROC-AUC score

Low Medium High Medium 0.65 0.97 – 1

Key Observations	model with linear decision boundary.	with feature importance insights.	but requires careful tuning.	boundaries with kernels.
Good baseline	Performs well	High accuracy	Effective for complex	

Output for Confusion Matrix



Discussion:

- **Overall Results:**
  1. Logistic Regression serves as a good baseline model that provides linear decision boundaries. It generally performs well with clean, linearly separable data.
  2. Random Forest demonstrates robust performance, benefiting from its ensemble nature by reducing variance and improving accuracy. It also provides insights into feature importance.
  3. XGBoost often delivers the highest accuracy due to its powerful boosting technique, which corrects errors from weak models in successive iterations. However, it requires careful tuning to avoid

overfitting.

4. Support Vector Classifier (SVC) works well for complex decision boundaries, especially when using different kernels. It performs well in high-dimensional feature spaces.

- **Overfitting and Underfitting Issues:**

1. Overfitting:

XGBoost and Random Forest may show signs of overfitting due to their complexity and ability to learn intricate patterns in the training data.

To mitigate overfitting, techniques like cross-validation, pruning, and limiting the depth of the trees have been employed.

Regularization parameters (like L1 and L2 penalties) have been applied in models like Logistic Regression and SVC to prevent them from fitting the noise in the data.

2. Underfitting:

Logistic Regression may suffer from underfitting, particularly if the data has non-linear patterns that it cannot capture with its linear decision boundary.

Using more complex models like SVC with non-linear kernels or ensemble methods can help in reducing underfitting by modeling more complex relationships.

- **Hyperparameter Tuning:**

  Hyperparameter tuning plays a crucial role in enhancing model performance, particularly for non-linear and ensemble methods.

  For Random Forest, parameters like the number of trees and maximum tree depth were tuned to balance the trade-off between bias and variance.

  XGBoost tuning involves adjusting parameters such as learning rate, maximum depth of the tree and the number of boosting rounds. Proper tuning significantly improved the accuracy and reduced overfitting tendencies.

  SVC involved tuning the `C` parameter and selecting the kernel type to achieve the best results for non-linear data.

- **Model Comparison and Model Selection:**

  Model Comparison:

  Random Forest also performed well due to its ability to handle both categorical and numerical data and its robustness to noisy data, though it required more computational resources.

  Logistic Regression served as a fast and interpretable model but lacked the complexity to handle non-linear relationships effectively.

SVC performed well in scenarios with non-linear relationships, particularly with the use of kernel methods, but was less interpretable compared to tree-based methods.

Model Selection:

The final model selection often involves a trade-off between interpretability, computational efficiency, and accuracy. For this project:

If interpretability is a priority (to understand feature importance and decision boundaries), Logistic Regression or Random Forest would be preferable.

If prediction accuracy and handling complex relationships are crucial, XGBoost would be the best choice due to its high performance.

Learning Outcome:

Google Colab Link - [project.ipynb]

Github Repository - https://github.com/sowmyadivi/ML-Project

Skills Used, Tools Used:

- **Skills Used:**
  - Data Preprocessing and Cleaning
  - Feature Engineering
  - Exploratory Data Analysis (EDA)
  - Machine Learning Model Implementation and Evaluation
  - Hyperparameter Tuning
  - Visualization Techniques
  - Problem-Solving in Data Science
  - Model Comparison and Selection

- **Tools Used:**
  - Programming Language: Python
  - Libraries: pandas, numpy, scikit-learn, XGBoost, matplotlib, seaborn
  - Development Environment: Jupyter Notebook, Google Colab
  - Version Control and Code Sharing: GitHub.

Dataset Used

The project used the **Customer Shopping Trends Dataset**, which includes features related to customer behavior and shopping patterns. This dataset was

processed to improve the quality of input for the machine learning models.

Customer Shopping Trends Dataset | Kaggle

What I Learned in This Project?

- **Data Processing Techniques:** Learned how to clean and preprocess data to enhance model performance.
- **Model Selection and Implementation:** Gained experience in implementing and comparing different machine learning algorithms like Logistic Regression, Random Forest, XGBoost, and SVC.
- **Hyperparameter Tuning:** Understood the importance of tuning model parameters to optimize accuracy and control overfitting.
- **Visualization and Interpretation:** Developed skills to create and interpret visual representations of data and model results.
- **Understanding of Overfitting and Underfitting:** Learned strategies to identify and mitigate these issues.
- **Project Collaboration:** Improved collaboration and code management skills using GitHub for version control.

Conclusion:

- This project successfully demonstrated the application of multiple

machine learning techniques to analyze and predict customer shopping trends using the Shopping Trends Dataset. Through data preprocessing, feature engineering, and implementing various classification models, we were able to gain insights into customer behavior and create a robust predictive framework.

- The use of ensemble methods, such as Random Forest and XGBoost, allowed us to capture complex patterns in the data, leading to improved model accuracy and performance. Additionally, the comparison of simpler models like Logistic Regression helped validate the effectiveness of more sophisticated approaches.

Accomplish?

- **Task (T):** The objective was to build a predictive model to analyze shopping trends, which was achieved by successfully implementing and tuning various machine learning algorithms.
- **Process (P):** The project followed a systematic approach, including data preprocessing, model selection, hyperparameter tuning, and evaluation. This structured methodology helped in extracting the most value from the data and fine-tuning the models for better results.
- **End Result (E):** The final models showed significant improvements in performance metrics, with XGBoost achieving the highest accuracy and predictive power.

Advantages and Limitations of Your Project:

Advantages:

- **Comprehensive Analysis:** The project employed multiple models to provide a thorough analysis of the data, allowing for better decision-making.
- **Model Comparison:** The systematic comparison of models helped identify the best-performing algorithm based on different evaluation metrics.
- **Scalability:** The methodology can be scaled to larger datasets or applied to similar customer trend analyzes with minor adjustments. ● **Interpretability:** Models like Logistic Regression and Random Forest provided insights into feature importance, aiding interpretability and decision-making.

Limitations:

- **Overfitting:** Some models, especially XGBoost, showed signs of overfitting to the training data, which required careful regularization and parameter tuning.
- **Computational Complexity:** Ensemble models like Random Forest and XGBoost were computationally more intensive and required longer training times, making them less suitable for real-time applications.
- **Data Quality:** The accuracy of the models heavily depended on the quality and completeness of the dataset. Missing values and noisy data might have impacted the predictions.
- **Non-Linear Relationships:** Logistic Regression struggled with non-linear patterns in the data, which limited its predictive power compared to other models.