Data Introduction

Background

- **Company Type**: International e-commerce company
- **Industry Focus**: Sells electronic products
- **Objective**: Discover key insights from customer database
- **Approach**: Use advanced machine learning techniques.

Project Objective

Apply classification models to answer key business questions:

- Can we predict customer satisfaction (rating)?
- Can we predict if a product is delivered on time?
- How do factors like Product Importance relate to high ratings or on-time deliveries?

Goal

- Enhance Customer Experience
- Optimize Operations
- Identify patterns that could increase business growth

Data Overview

Source of Data:

- Kaggle: Customer Analytics Dataset

Context:

- International e-commerce company specializing in electronics.
- Analyzing shipment delivery and customer interactions.

Project Objective:

- Target variable: "Reached.on.Time_Y.N"
  - Class 1 ("Reached") than class 0 ("Not Reached").
- Classification models applied to predict delivery status and extract business insights.

Variable Description

Variables Name	Description
ID	Unique ID for customer
Warehouse_block	Warehouse A,B,C,D,F
Mode_of_Shipment	Ship, Flight, Road
Customer_care_calls	# of care calls made
Customer_rating	Rating give (1-5)
Cost_of_the_Product	Price of Product (USD)

Variables Name	Description
Prior_purchases	# of prior purchases
Product_importance	Low, Medium, High importance
Gender	M/F
Discount_offered	Discount given
Weights_in_gms	Product weight
Reached.om.Time_Y.N.	1 = Delayed, 0 = Delivered

Data Cleaning Process

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10999 entries, 0 to 10998
Data columns (total 12 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
```

"The dataset has 10,999 records with no missing values in key numerical fields."

```
# Generate descriptive statistics for numerical columns
df.describe()
```

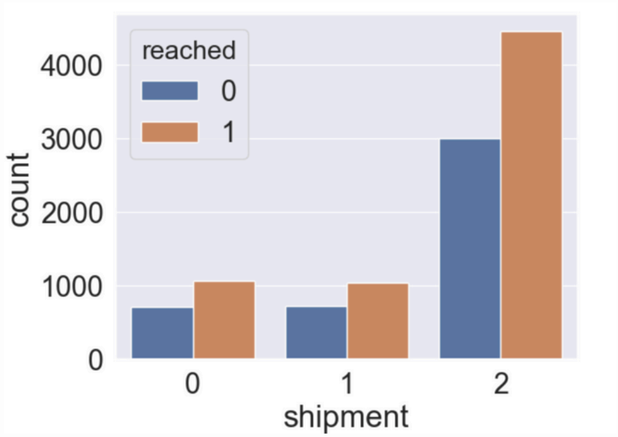
	id	calls	rating	price	purchases	discount	weight	reached
count	10999.00000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000
mean	5500.00000	4.054459	2.990545	210.196836	3.567597	13.373216	3634.016729	0.596691
std	3175.28214	1.141490	1.413603	48.063272	1.522860	16.205527	1635.377251	0.490584
min	1.00000	2.000000	1.000000	96.000000	2.000000	1.000000	1001.000000	0.000000
25%	2750.50000	3.000000	2.000000	169.000000	3.000000	4.000000	1839.500000	0.000000
50%	5500.00000	4.000000	3.000000	214.000000	3.000000	7.000000	4149.000000	1.000000
75%	8249.50000	5.000000	4.000000	251.000000	4.000000	10.000000	5050.000000	1.000000
max	10999.00000	7.000000	5.000000	310.000000	10.000000	65.000000	7846.000000	1.000000

“Descriptive analysis for columns”

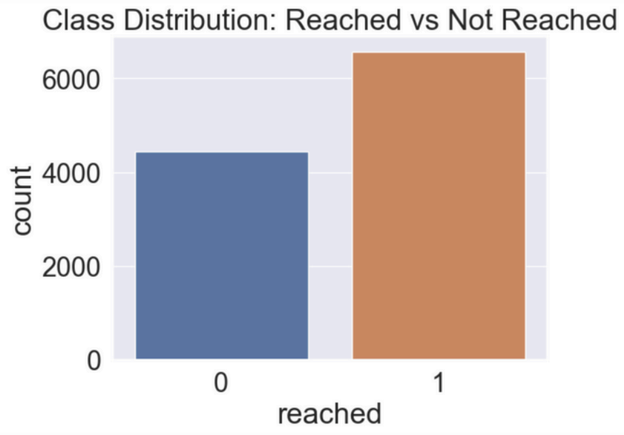
Data Visualization

Purchase per warehouse	Shipment count per warehouse	Avg discount used
3.6020	1834	44.0
3.5480	3666	59.0
3.5777	1833	48.0
3.5750	1833	10.0
3.5548	1833	46.0

- "Warehouse 2 processes the highest number of shipments (3666) with the highest average discount (59%)."
- "Purchase rates are fairly consistent across warehouses, averaging around 3.55–3.60 purchases."
- "Warehouse 3 offers the lowest average discount (10%) among all warehouses."

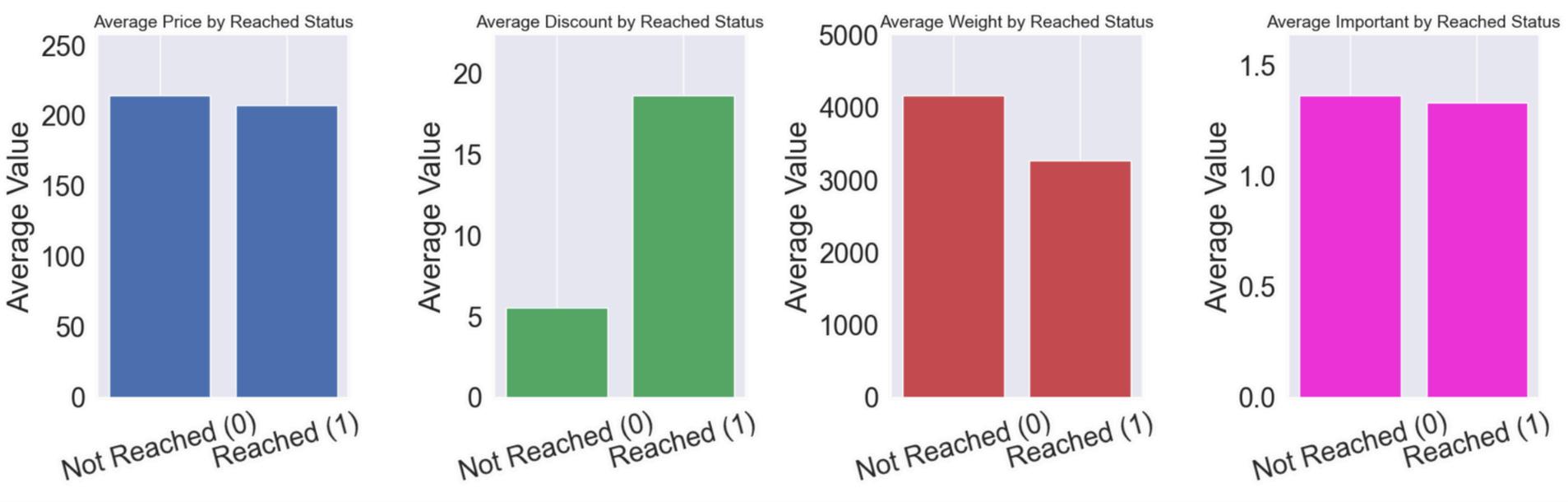


- "Shipment mode 2 has the highest number of successful and failed deliveries."
- "Shipment modes 0 and 1 have relatively balanced delivery outcomes."



- "Around 60% of shipments were delivered successfully, while 40% faced delays."

Data Visualization



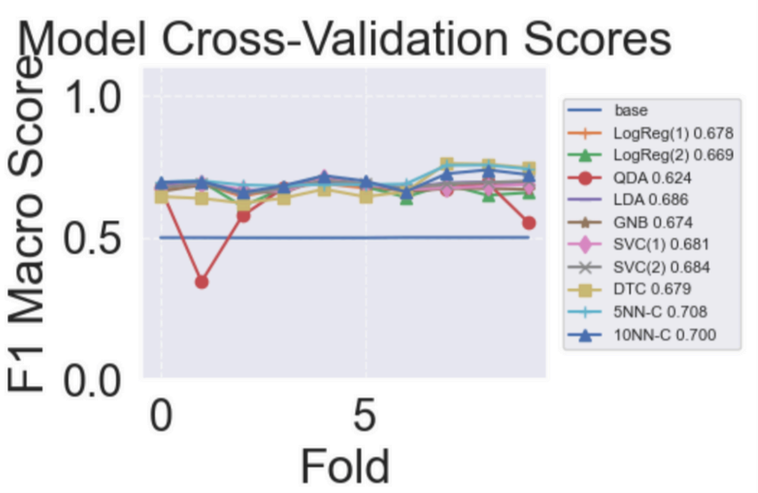
Standardization

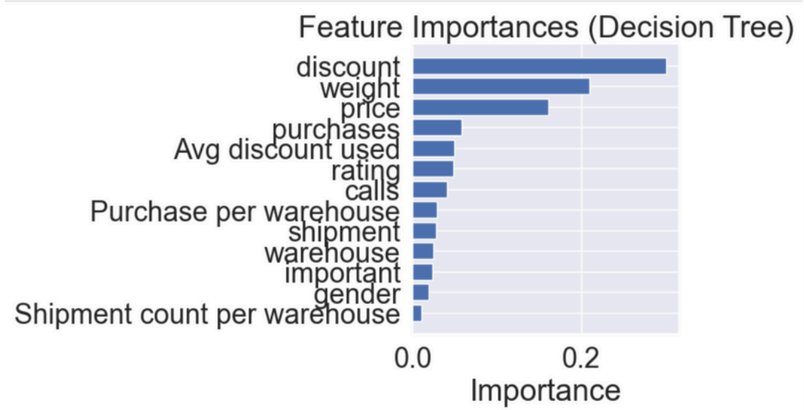
- Standardized all numerical features for consistency
- Shown here: price column (first 1000 rows only)
- Used StandardScaler to center data (mean = 0, std = 1)
- Ensures fair comparison across features

Technology

python™

& Machine Learning Approach





Classification — Ran the classifier parade (10 models) and selected the two models with the highest performance for our dataset

KNN & DTC — Based on F1 Macro Score, KNN achieved the highest score (0.708), and DTC showed strong performance as well (0.679)

SMOTE — Was used to address the imbalance between the target classes by synthetically generating new examples of the minority class. This avoids bias toward the majority class

```
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X_train, y_train)
```

Analytical Method

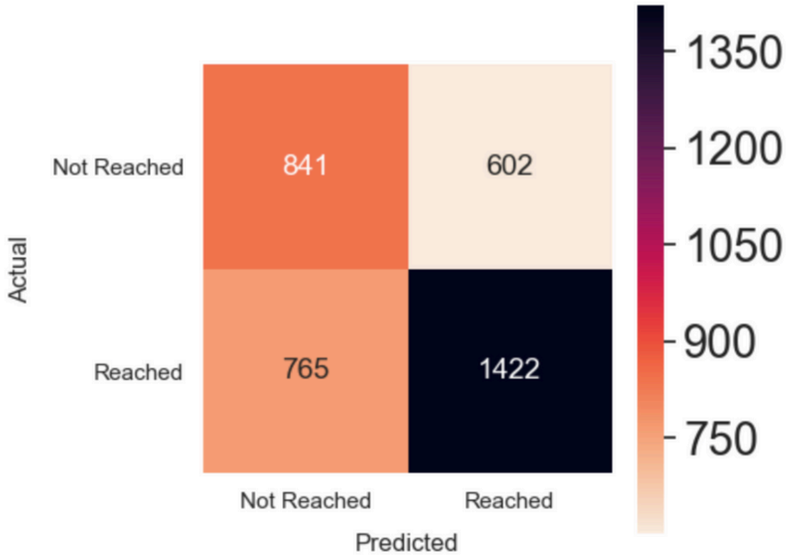
KNeighborsClassifier

Tuning Hyperparameters

GridSearchCV found k=1 had the highest test score (0.746), capturing local patterns after SMOTE. However, perfect training accuracy suggests overfitting

Multiple Hyperparameters

Choosing n_neighbors=2 and distance weighting helps capture local patterns while reducing the impact of distant, less relevant points



62% MODEL ACCURACY

70% OF PREDICTED 1S WERE ACTUALLY 1

THE MODEL PERFORMS BETTER AT PREDICTING CLASS 1 ("REACHED") THAN CLASS 0 ("NOT REACHED")

	precision	recall	f1-score	support
0	0.52	0.58	0.55	1443
1	0.70	0.65	0.68	2187
accuracy			0.62	3630
macro avg	0.61	0.62	0.61	3630
weighted avg	0.63	0.62	0.63	3630

Analytical Method

DTC

Training Strategy

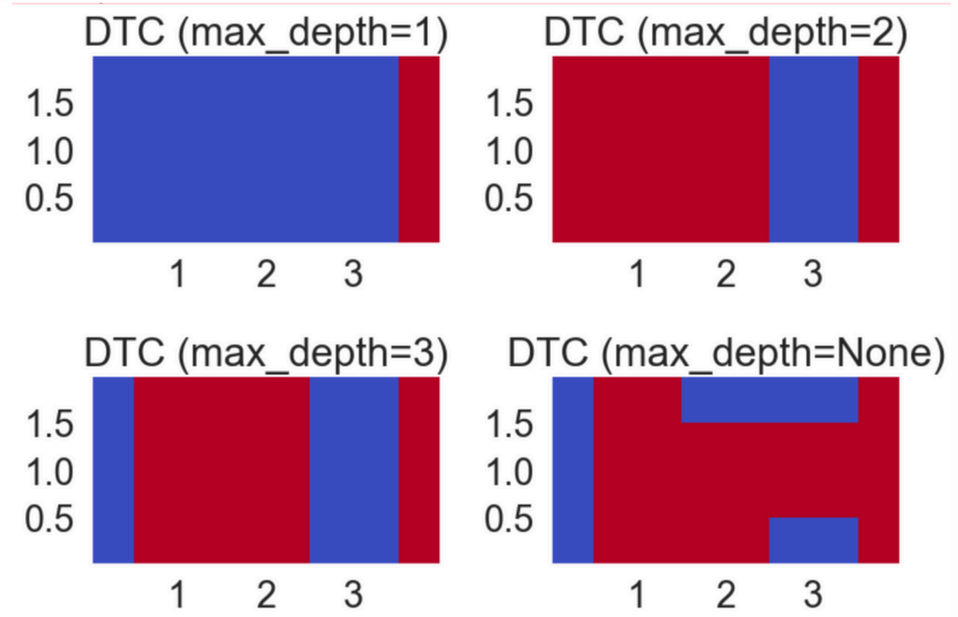
3-Fold Cross-Validation was used after applying SMOTE to balance the data. Model was tuned with max_depth=3 to avoid overfitting.

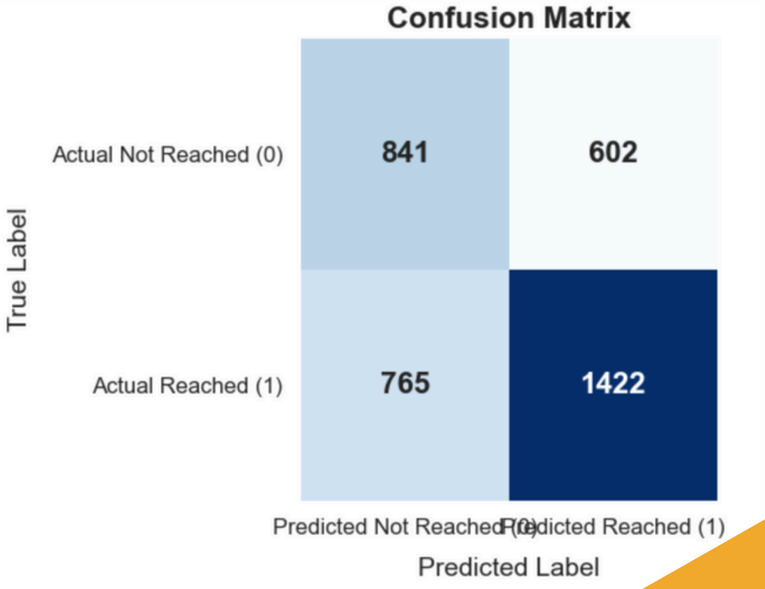
Tree Behavior

First split based on discount feature. Later splits based on weight, purchases, and average discount used.

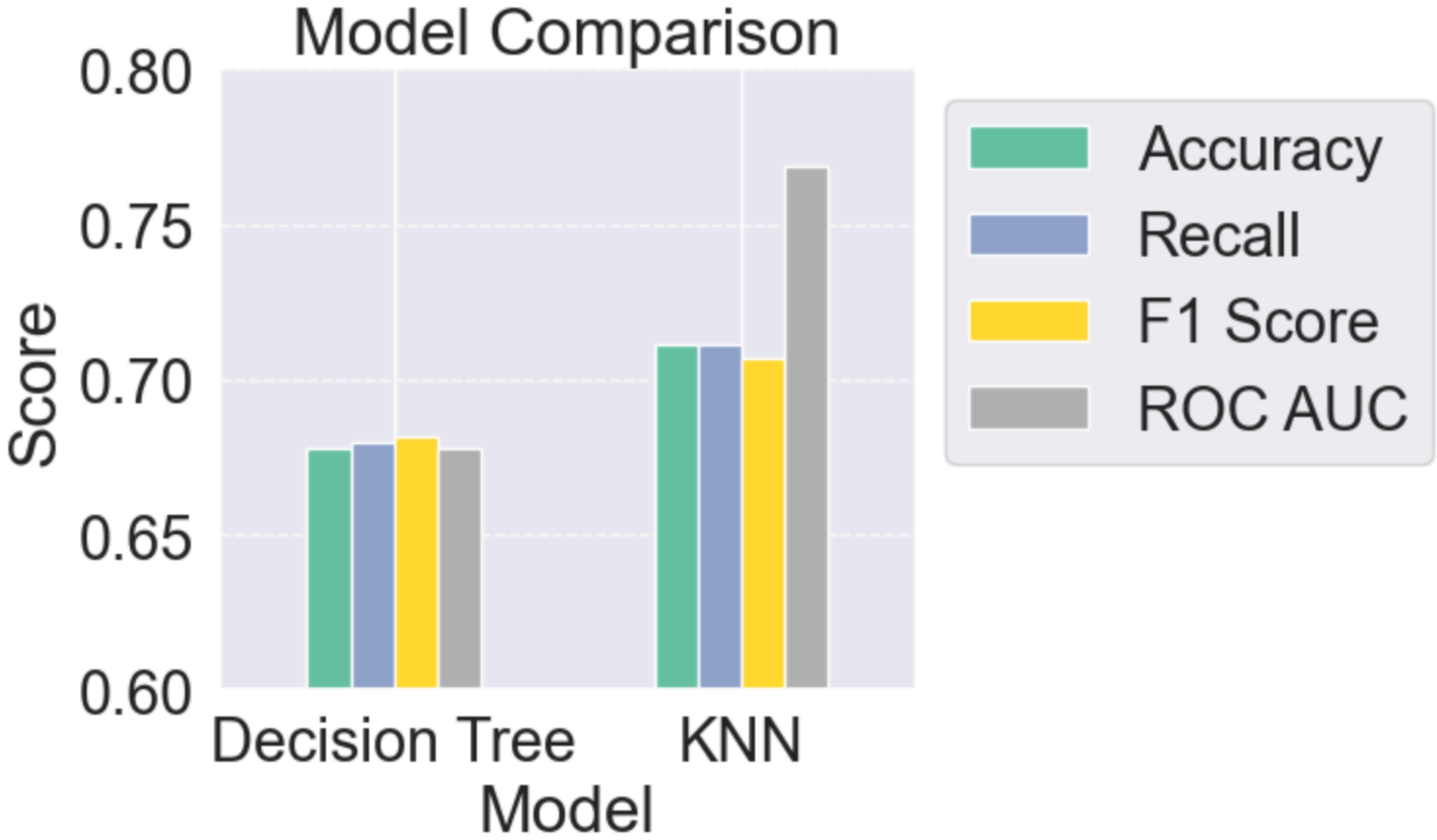
67% MODEL ACCURACY

72% OF PREDICTED "REACHED" WERE ACTUALLY CORRECT





Model Performance Comparison



Technical Challenges

Unbalanced Dataset
Required applying SMOTE to create a balanced training set

Plot & visualization Issues
Legends overlapping plots and boundary visualization failures

Overfitting risk in KNN
k=1 achieved perfect training accuracy but risked memorization

Nested Cross-validation problems
Performance worsened compared to simple cross-validation

Standardization Issue
Due to 10K+ rows crashing the computer, only the first 1,000 rows were visualized

Data
Business Insights

Insights from the Data

KNN (k=2, distance-based) outperformed other models after SMOTE resampling

Class 1 (reached) showed stronger predictive patterns than Class 0

Impact & Value to Business

Improves campaign efficiency by focusing on high-probability leads

Supports data-driven decision-making for resource allocation

Recommendation & Conclusion

Summary:

- Higher discounts and heavier products are linked to delayed deliveries.
- Shipments via road transport have a higher delay rate.

Recommendations:

- Optimize logistics for heavy and discounted products.
- Focus on improving road shipping processes.
- Continue monitoring and predicting shipment outcomes using machine learning models.

Real-world applications:

- Enhanced delivery performance and customer satisfaction.
- Reduced operational costs through proactive shipment management.

Data References

Gopalani, P. (2021, February 23). E-commerce shipping data. Kaggle. Link

pandas.DataFrame.reset_index. Pandas 2.2.3 Documentation. Link

SMOTE - Synthetic Minority Oversampling Technique. Imbalanced-Learn 0.13.0 Documentation. Link

plot_tree Function. Scikit-learn Documentation. Link