Predicting Customer Response to Gold Membership Offer

Bracha Stein

07.13.2025

Agenda

1. **Project Overview**
   - Business problem
   - Dataset & features
   - Objective & target
2. **Modeling Approach**
   - Data preparation
   - Model comparison
   - Final model selection (XGBoost)
   - Model performance & interpretation (SHAP)
3. **Business Application**
   - Strategic insights
   - Recommended actions
   - Next steps

Business Problem & Objective

Superstore is offering a discounted $499 Gold Membership.

Objective: Predict which customers are likely to accept the offer.

Helps improve ROI by focusing outreach on high-likelihood responders.

Business Value

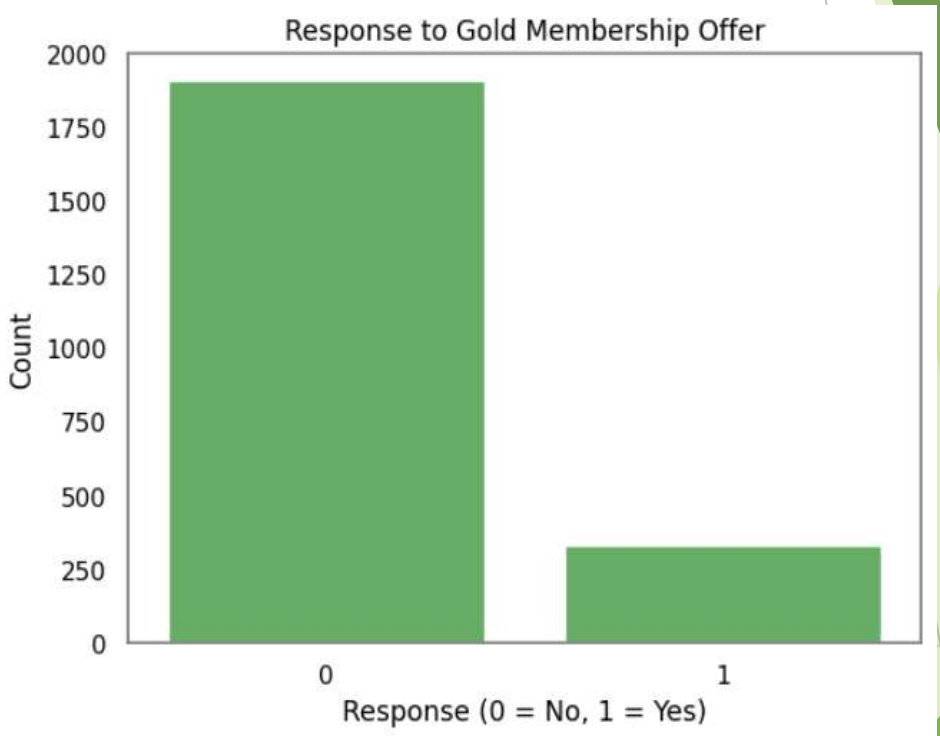
- **Cost Efficiency:** Reduces wasted outreach
- **Higher Conversion Rates:** Precise targeting increases sign-ups
- **Customer Insight:** Segments customers by likelihood and traits
- **Scalability:** Model can be reused across future campaigns
- **Estimated ROI**: Targeting top 30% of likely responders could reduce costs by ~40% while capturing most conversions

Data Overview

- Records: 2,240 customer profiles
- Target Variable: Response (0 = Declined, 1 = Accepted)
- Predictors: Recency (days since last purchase), Income, Spending (Wines, Meat, Gold), Tenure, Education, Marital Status, Web Visits
- Class Imbalance: Only 15% of customers accepted the offer

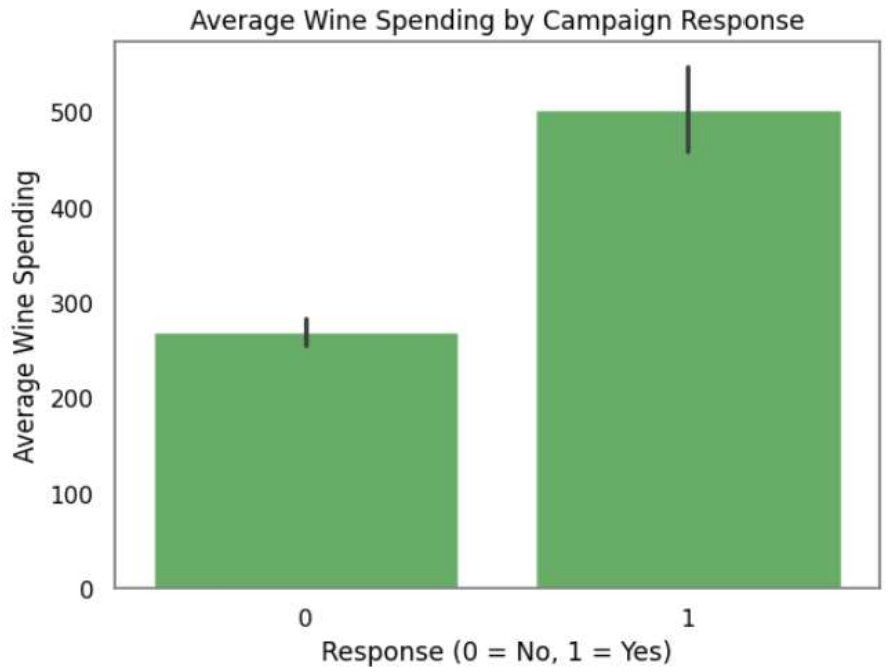
EDA – Class Imbalance & Recency

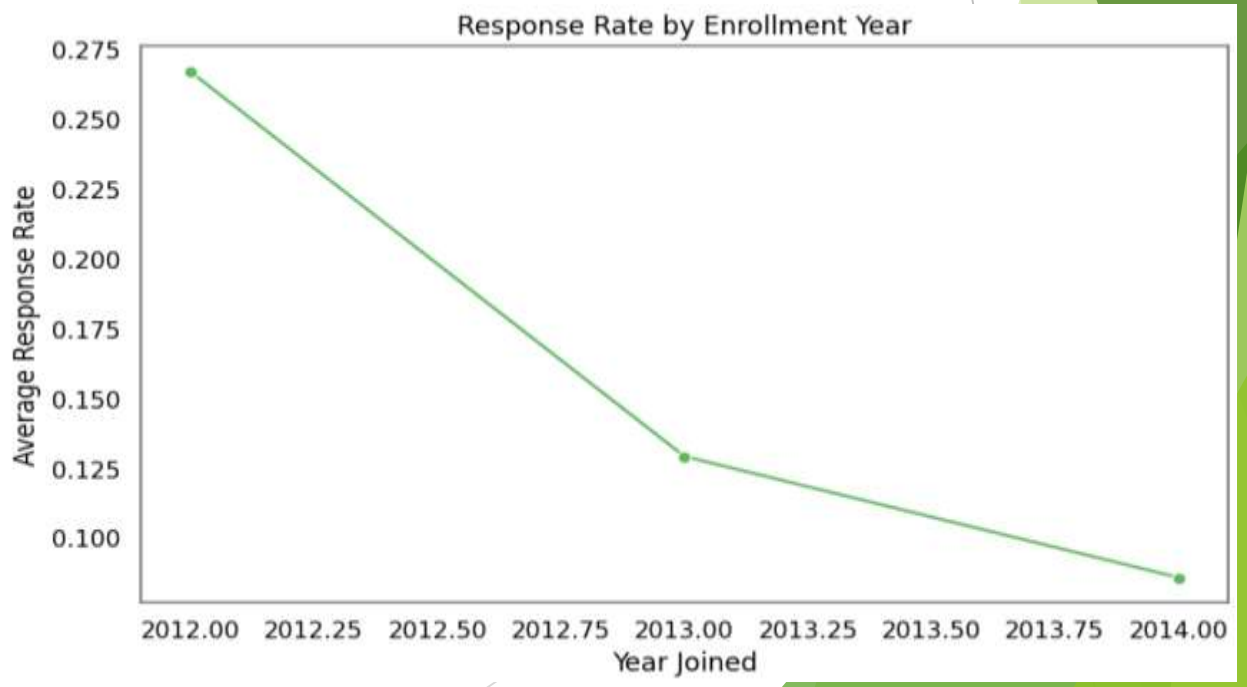
- Class distribution: 85% No, 15% Yes → Imbalance handled with class weights
- Recency skew: Most customers purchased recently, few were inactive
- High Recency (lapsed buyers) correlated with higher response likelihood



EDA – Spending & Demographics

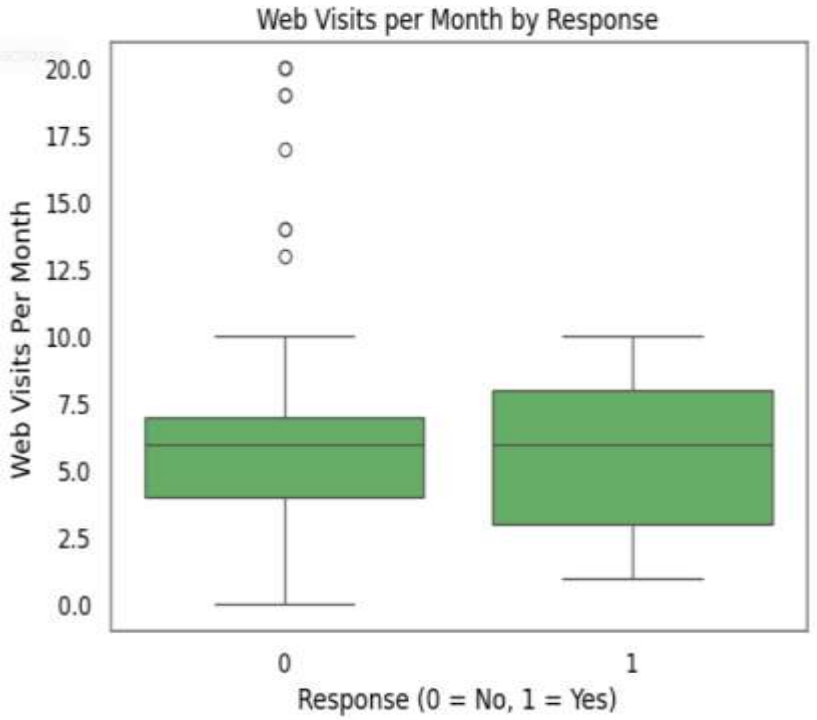
- Higher spending on wines, meat, gold = more likely to respond
- Longer-tenured customers (earlier join year) = higher response rate
- Families with teenagers respond more; those with young children less likely

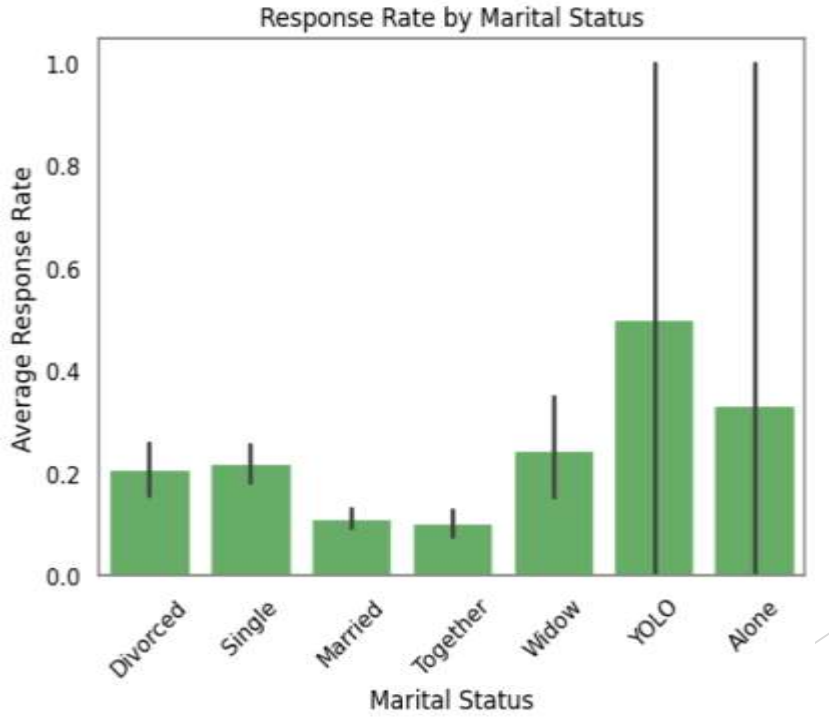


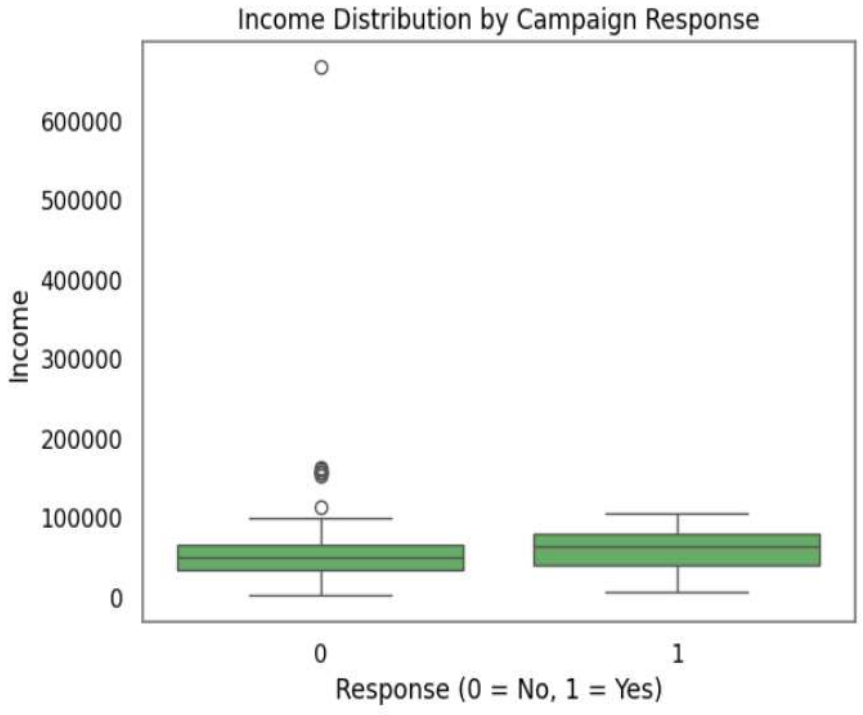


EDA – Web Visits & Marital Status

- Lower web visits per month = higher campaign responsiveness
- Response varies by marital status: Married & Together > Single & Divorced
- Income boxplots show wider distribution among responders







High-Level Approach

Steps taken:

- Cleaned and prepared data (missing income, encoding, tenure)
- Extracted join year from customer signup date (used as loyalty indicator)
- Tested 3 models: Logistic Regression, Random Forest, XGBoost
- Evaluated with accuracy, precision, recall, F1, ROC AUC
- Interpreted results with SHAP

Model Comparison

Model	Accuracy	Precision (1)	Recall (1)	F1 Score	ROC AUC
XGBoost	0.88	0.60	0.48	0.53	0.88
Logistic Regression	0.75	0.35	0.81	0.49	0.86
Random Forest	0.87	0.68	0.23	0.34	0.88

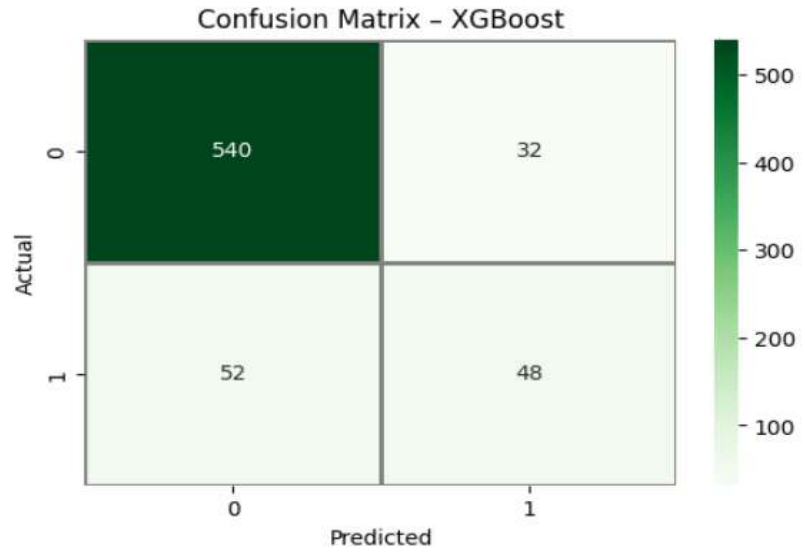
Model Selection:

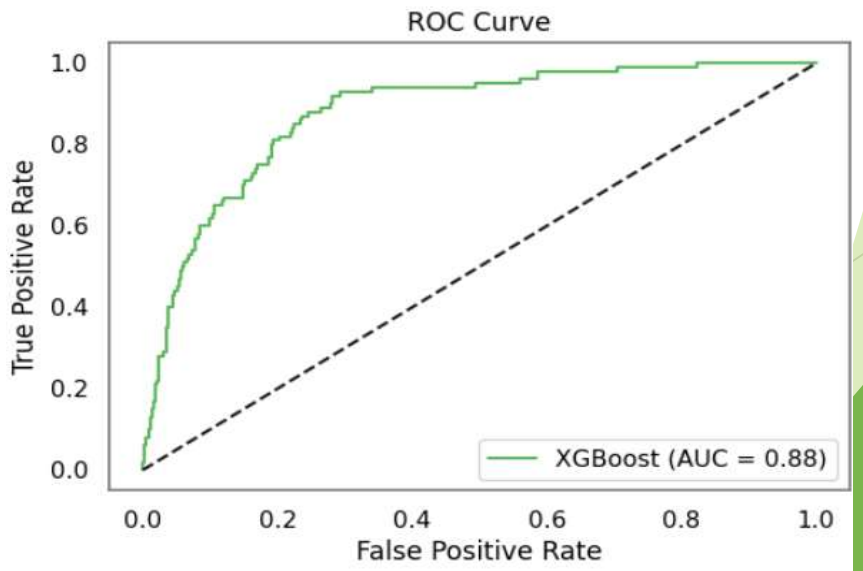
- XGBoost offers best balance of recall + precision
- Logistic: Over-predicts responders → costly
- Random Forest: High precision, low recall → misses too many responders

Final Model Performance* (XGBoost)

- Accuracy: 88%
- Precision: 60%
- Recall: 48%
- ROC AUC: 0.88
- Confusion Matrix:
  - TP: 48, FN: 52, TN: 540, FP: 32
- Balanced results: manages campaign cost while capturing nearly half of responders

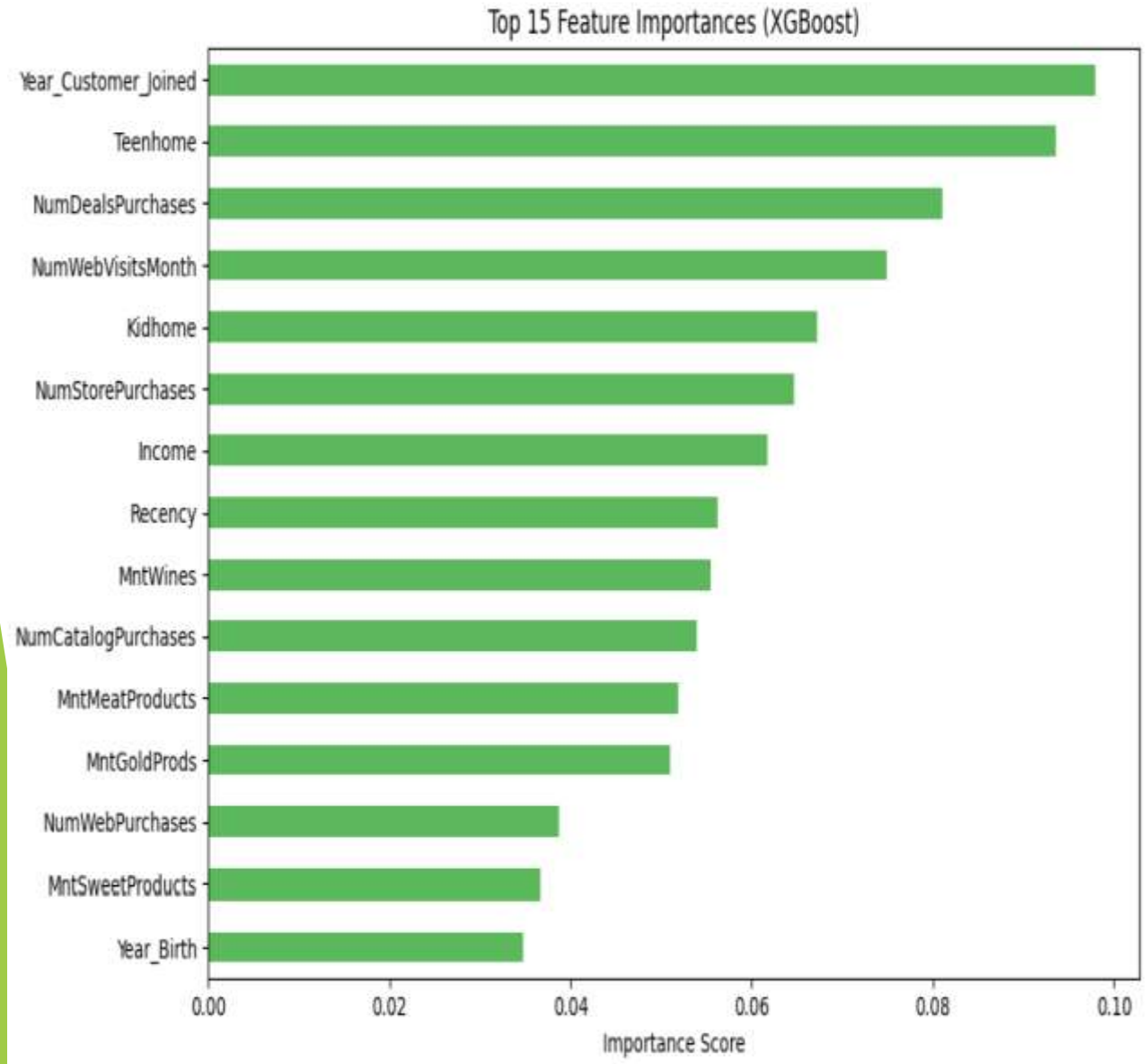
*All metrics shown are based on performance on the held-out 30% test set (stratified sampling)

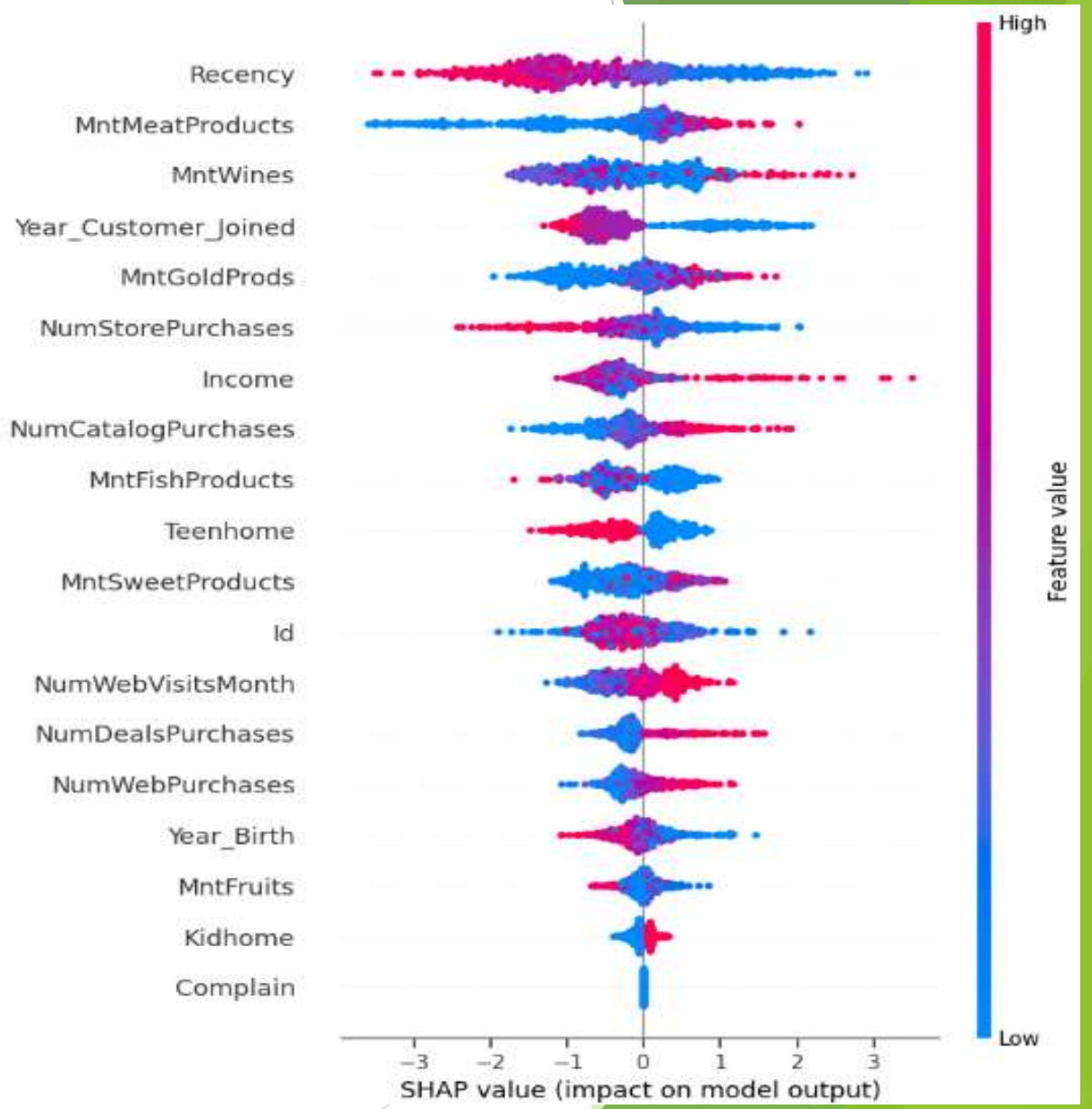




Shap Insights – Key Features

- Top Predictors:
  - Recency: Long gaps → higher acceptance
  - MntWines, MntMeatProducts, MntGoldProds → premium spenders
  - Earlier Join Year: Indicates longer loyalty and correlates with higher response rates
  - Teenhome vs Kidhome: Teens ↑, small children ↓ response rate
- SHAP Summary: Confirms business intuition and shows local/global feature impact


Top 15 Feature Importances (XGBoost)
Year_Customer_Joined
Teenhome
NumDealsPurchases
NumWebVisitsMonth
Kidhome
NumStorePurchases
Income
Recency
MntWines
NumCatalogPurchases
MntMeatProducts
MntGoldProds
NumWebPurchases
MntSweetProducts
Year_Birth
0.00
0.02
0.04
0.06
0.08
0.10
Importance Score


Recency
MntMeatProducts
MntWines
Year_Customer_Joined
MntGoldProds
NumStorePurchases
Income
NumCatalogPurchases
MntFishProducts
Teenhome
MntSweetProducts
Id
NumWebVisitsMonth
NumDealsPurchases
NumWebPurchases
Year_Birth
MntFruits
Kidhome
Complain
−3
−2
−1
0
1
2
3
SHAP value (impact on model output)
High
Low
Feature value

Business Strategy & Implementation

- Integrate model into CRM for automated scoring
- Run A/B tests: Model-based vs traditional targeting
- Use response probability thresholds to tune reach (cost vs conversion)
- Tailor messages: Lapsed premium buyers vs long-term loyalists
- Target by segment: Wine/meat spenders, low web activity, etc.

Next Steps

- Hyperparameter tuning for XGBoost
- Deploy for real-time campaign targeting
- Add new features: past campaign history, website behavior
- Monitor monthly → retrain as behaviors shift
- Layer in Customer Lifetime Value for deeper prioritization

Appendix

- Data Preparation & Feature Engineering:

```
# Convert join date to datetime format
df['Dt_Customer'] = pd.to_datetime(df['Dt_Customer'])


# Fill missing income values with median
df['Income'] = df['Income'].fillna(df['Income'].median())


# One-hot encode Education and Marital_Status
df = pd.get_dummies(df, columns=['Education', 'Marital_Status'], drop_first=True)


# Extract Year_Customer_Joined (used as a proxy for tenure)
df['Year_Customer_Joined'] = df['Dt_Customer'].dt.year
```

- Model Training – Train/Test Split & XGBoost:

```
# Define X and y
y = df["Response"]
X = df.select_dtypes(include=np.number).drop("Response", axis=1)

# 1. Import XGBoost
import xgboost as xgb

# 2. Split into training and test sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.3, random_state=42)


# 3. Train the XGBoost model
model = xgb.XGBClassifier(use_label_encoder=False, eval_metric='logloss', random_state=42)
model.fit(X_train, y_train)

# 4. Plot feature importances
import pandas as pd
import matplotlib.pyplot as plt

importances = pd.Series(model.feature_importances_, index=X.columns)
top_features = importances.sort_values(ascending=False).head(15)

plt.figure(figsize=(10, 6))
top_features.plot(kind='barh')
plt.title("Top 15 Feature Importances (XGBoost)")
plt.xlabel("Importance Score")
plt.gca().invert_yaxis()
plt.tight_layout()
plt.show()
```

- SHAP Model Explanation:

```
import shap
explainer = shap.TreeExplainer(model)
shap_values = explainer.shap_values(X_test)

# Summary plot (interactive)
shap.summary_plot(shap_values, X_test)
```

- Model Evaluation – Classification, AUC, Confusion Matrix, ROC Curve:

```
from sklearn.metrics import classification_report, confusion_matrix, roc_auc_score, roc_curve
import seaborn as sns

# Predict on test data
y_pred = model.predict(X_test)
y_prob = model.predict_proba(X_test)[:, 1]

# Classification report
print("Classification Report:\n")
print(classification_report(y_test, y_pred))

# ROC AUC score
roc_auc = roc_auc_score(y_test, y_prob)
print(f"ROC AUC Score: {roc_auc:.4f}")

# Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# ROC Curve
fpr, tpr, _ = roc_curve(y_test, y_prob)
plt.figure(figsize=(6, 4))
plt.plot(fpr, tpr, label=f"XGBoost (AUC = {roc_auc:.2f})")
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve")
plt.legend()
plt.show()
```

- Comparison Models – Logistic & Random Forest

```
# Logistic Regression (scaled pipeline)
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import make_pipeline

log_model = make_pipeline(
    StandardScaler(),
    LogisticRegression(class_weight='balanced', max_iter=1000)
)
log_model.fit(X_train, y_train)

# Random Forest
from sklearn.ensemble import RandomForestClassifier

rf_model = RandomForestClassifier(n_estimators=100, class_weight='balanced',
random_state=42)
rf_model.fit(X_train, y_train)
```