**PREDICTING CUSTOMER LIFETIME VALUE (CLTV) IN**

**E-COMMERCE**

**1. Introduction**

This project aims to predict Customer Lifetime Value (CLTV) for an e-commerce company using historical customer data. Accurate CLTV prediction helps businesses in customer segmentation, marketing budget allocation, and customer retention strategies. Understanding CLTV allows companies to identify high-value customers and tailor their marketing strategies accordingly.

**2. Data Exploration and Preprocessing**

**2.1 Loading the Dataset**

The dataset was loaded using Pandas, and initial exploration was performed to understand its structure. The dataset contains features such as CustomerID, Gender, Location, Website Engagement, and various numerical features.

**import pandas as pd**

**data = pd.read\_csv('dataset.csv')**

**print(data.info())**

**2.2 Handling Missing Values**

Missing values were checked and handled appropriately. Any missing values were imputed using the mean for numerical features and the mode for categorical features.

**data.fillna(data.mean(), inplace=True)**

**2.3 Categorical Encoding**

Categorical variables (Gender, Location, Website Engagement) were converted into numerical format using One-Hot Encoding to ensure compatibility with the machine learning models.

**from sklearn.preprocessing import OneHotEncoder**

**categorical\_features = ['Gender', 'Location', 'WebsiteEngagement']**

**encoder = OneHotEncoder(handle\_unknown='ignore')**

**encoded\_features = encoder.fit\_transform(data[categorical\_features]).toarray()**

**2.4 Feature Scaling**

Numerical features were scaled using Standard Scaler to standardize the values, which helps many models perform better by removing bias due to different scales of features.

**from sklearn.preprocessing import StandardScaler**

**numerical\_features = ['Age', 'AnnualIncome', 'PurchaseFrequency', 'AverageOrderValue', 'TotalPurchases', 'CustomerTenure', 'CustomerServiceInteractions', 'MarketingSpend']**

**scaler = StandardScaler()**

**data[numerical\_features] = scaler.fit\_transform(data[numerical\_features])**

**3. Feature Engineering**

**3.1 Creating New Features**

**To enrich the dataset, new features were created:**

* Average Purchase Value: Calculated by dividing Total Purchases by Purchase Frequency.
* Engagement Score: A combination of Website Engagement and Purchase Frequency to measure user involvement.

**data['AvgPurchaseValue'] = data['TotalPurchases'] / data['PurchaseFrequency']**

**data['EngagementScore'] = data['WebsiteEngagement'] \* data['PurchaseFrequency']**

**3.2 Dimensionality Reduction**

Although the number of features was manageable, Principal Component Analysis (PCA) can be applied to reduce the dimensionality if necessary.

**from sklearn.decomposition import PCA**

**pca = PCA(n\_components=5)**

**4. Model Building**

**4.1 Splitting the Dataset**

The dataset was split into an 80-20 ratio for training and testing. This ensures that the model is evaluated on unseen data, giving a better idea of its real-world performance.

**from sklearn.model\_selection import train\_test\_split**

**X = data.drop(columns=['CLTV']) # Replace 'CLTV' with your target column name**

**y = data['CLTV'] # Replace 'CLTV' with your target column name**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)**

**4.2 Model Selection and Hyperparameter Tuning**

Several regression models were considered, including Linear Regression, Decision Trees, Random Forest, and Gradient Boosting models like XGBoost. After evaluating performance, the XGBoost model was selected due to its ability to handle complex non-linear relationships effectively. Hyperparameter tuning was done using Randomized Search to optimize the model’s performance.

**from sklearn.model\_selection import RandomizedSearchCV**

**from xgboost import XGBRegressor**

**pipeline = Pipeline(steps=[('preprocessor', preprocessor), ('model', XGBRegressor())])**

**param\_dist = {**

**'model\_\_n\_estimators': [100, 200, 300],**

**'model\_\_max\_depth': [3, 5, 7, 9],**

**'model\_\_learning\_rate': [0.01, 0.1, 0.2],**

**'model\_\_subsample': [0.5, 0.8, 1.0],**

**'model\_\_colsample\_bytree': [0.5, 0.8, 1.0],**

**'model\_\_min\_child\_weight': [1, 2, 3]**

**}**

**random\_search\_xgb = RandomizedSearchCV(pipeline, param\_distributions=param\_dist, n\_iter=50, cv=3, random\_state=42, scoring='neg\_mean\_squared\_error')**

**random\_search\_xgb.fit(X\_train, y\_train)**

**best\_params = random\_search\_xgb.best\_params\_**

**print("Best parameters found:", best\_params)**

**5. Model Evaluation**

**5.1 Evaluation Metrics**

**The performance of the model was evaluated using the following metrics:**

* Root Mean Squared Error (RMSE): Measures the average magnitude of errors.
* Mean Absolute Error (MAE): Measures the average absolute difference between actual and predicted values.
* R-squared: Explains how well the variance in the data is explained by the model.

**from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score**

**y\_pred = random\_search\_xgb.predict(X\_test)**

**mse = mean\_squared\_error(y\_test, y\_pred)**

**mae = mean\_absolute\_error(y\_test, y\_pred)**

**rmse = np.sqrt(mse)**

**r2 = r2\_score(y\_test, y\_pred)**

**relative\_rmse = (rmse / y\_test.mean()) \* 100**

**5.2 Feature Importance**

Feature importance analysis was performed to understand which features most significantly contribute to predicting CLTV. XGBoost provides built-in functionality for visualizing feature importance.

**import matplotlib.pyplot as plt**

**from xgboost import plot\_importance**

**plot\_importance(random\_search\_xgb.best\_estimator\_.named\_steps['model'])**

**plt.show()**

**6. Insights and Recommendations**

**6.1 Key Insights**

* Marketing Spend: A significant factor influencing CLTV. Optimizing marketing spend may lead to higher customer retention.
* Customer Tenure: Longer tenure customers tend to have higher CLTV, indicating the importance of retaining long-term customers.
* Average Order Value: Positively correlated with CLTV, implying that efforts to increase order size (e.g., through promotions) may improve CLTV.

**6.2 Actionable Recommendations**

* Customer Segmentation: Focus on high-tenure customers with personalized offers to increase their average order value and overall CLTV.
* Marketing Campaigns: Invest more in campaigns targeting customers who have higher website engagement and purchase frequency.
* Customer Service: Improve customer service interactions, as they play a significant role in retention.