What is Customer Segmentation?

Customer segmentation is the act of grouping your audience into sub-categories based on similarities in user profiles. Segmentation allows you to find the best content, channel, and timing for your campaigns to be sent to potential leads or current customers. It empowers you to provide personalized experiences along each touch point across the user journey.

Segmentation is the process of dividing your customers up into different groups, with each group sharing similar characteristics, to improve engagement, sales and loyalty.

A key focus in marketing is to build relationships with profitable customers. Marketing is not just about a single purchase, a single conversion – the goal is to grow your customer base and to build long term value.

Understanding customers, segmenting them and personalizing marketing campaigns, offers and communications is part of what is broadly known as Customer Relationship Management (CRM).

According to Harvard Business Review, over 30,000 new consumer products are launched each year.

And 95% of them fail for one of these seven reasons:

- ***Failure to understand consumer needs and wants***.
- Fixing a non-existent problem.
- ***Targeting the wrong market***.
- Incorrect pricing.
- Weak team and internal capabilities.
- Prolonged development or delayed market entry.
- Poor execution.

Customer Segmentation Benefits

- helps to detect and exploit new market opportunities.
- improves how to predict customer behavior.
- Increased customer retention and loyalty.
- improves the perception of a brand through personalization.
- streamlines and improves workflow.
- helps to improve customer lifetime value.
- Email marketers have witnessed a 760% increase in revenue by segmenting their email campaigns.

Types of Customer Segmentmentation

- Geographical segmentation :
    - Countries, Region, States
- Market segmentation
    - High margin market
    - Low profit market
- Behavioral Segmentation
    - High margin
    - Most profitable
    - Loyalist
- Demographic segmentation
    - Age
    - Gender
    - Device
    - Interest

How To Do Customer Segmentation using Google Analytics Data

Google Analytics allows you to create detailed customer segments. You can find plenty of demographic statistics under the *audience* tab, as well as some information about the (other) interests of your visitors. Click “add segment” to add any segment you’d like and keep track of…

Different Analysis Techniques for Customer Segmentation in Python

1. **RFM Analysis**: RFM (Recency, Frequency, Monetary) analysis is a technique that evaluates customers based on their recent purchase behavior, frequency of purchases, and monetary value. It segments customers into groups such as high-value, low-value, frequent purchasers, or dormant customers. Descriptive and exploratory technique rather than predictive
2. **Cluster Analysis or Modeling**: Cluster analysis, often implemented using techniques like k-means clustering, groups customers based on similarities in their attributes or behaviors. The goal of clustering is to discover hidden patterns or structures in the data and create meaningful groups or clusters. It aims to identify distinct segments by minimizing the differences within each segment and maximizing the differences between segments. descriptive or predictive
3. **Market Basket Analysis**: Market basket analysis examines customers' purchasing patterns to identify associations between products frequently purchased together. It helps uncover cross-selling opportunities and segment customers based on their product preferences and affinities. Descriptive and exploratory technique rather than predictive
4. **Social Network Analysis**: Social network analysis examines the relationships and interactions among customers to identify influential customers or groups and their impact on purchasing decisions. It can be useful for

segmenting customers based on their social connections and influence. Descriptive and exploratory technique rather than predictive

5. **Cohort Analysis**: Cohort analysis groups customers based on shared characteristics or behaviors within a specific timeframe. Cohort analysis typically involves creating cohorts based on specific criteria, such as the time of acquisition, user behavior, or demographic attributes. The cohorts are then tracked and analyzed to understand how their behavior or metrics change over time. Descriptive and exploratory technique rather than predictive

Machine Learning Models for Customer Segmentation

Clustering Model: Using K-means algorithm Algorithm

1. **Data Preparation**: Gather relevant customer data, such as demographics, purchase history, behavioral data, or any other variables that might be useful for segmentation. Ensure that the data is preprocessed, cleaned, and appropriately scaled.
2. **Determine the Number of Segments** (k): Decide on the desired number of customer segments. This can be based on domain knowledge, business requirements, or by using techniques like the elbow method or silhouette analysis to determine the optimal number of clusters.
3. **Apply the k-means Algorithm**: Use the k-means algorithm to cluster the customers into k segments. The algorithm iteratively assigns each customer to the nearest cluster centroid (mean) based on a distance measure (usually Euclidean distance) and recalculates the centroids until convergence.
4. **Assign Customers to Segments**: Once the k-means algorithm has converged, each customer will be assigned to one of the k segments based on their proximity to the cluster centroids. Customers belonging to the same segment are expected to have similar characteristics or behaviors.

Propensity Scoring Model: Using Logistics regression

- Logistics regression is a popular algorithm for propensity modeling. It models the relationship between the predictor variables and the binary outcome variable using a logistic function, which estimates the probability of belonging to a specific class. Propensity scoring models can be used for customer segmentation by predicting the probability of customers belonging to a specific segment or exhibiting a particular behavior.
- Case study: Create a segment of leads with high probability of sale and want to build a model to prioritize sales calls to online potential customers (website users), based on store visit likelihood
- Data: site visitor data from GA, have CRM data that tracks leads
    - GA data have dimension: Geo, time device tye
    - Details on whether they'd turn into a sales or not
    - Solution: use logistics regression. Build the model with logistics regression to score future leads.
    - End solution: Take new leads that come in, plug into the model and the model will output the new need likelihood of becoming a conversion. A system will take a high probability of sales leads and follow up with them right away. E.g prioritize calls if probability > 0.5

The process typically involves the following steps:

1. **Building the Propensity Model**: First, a propensity model is constructed using historical data with labeled outcomes. The model is trained on a set of predictor variables (such as demographics, purchase history, website interactions, etc.) and the corresponding binary outcome variable (segment membership or behavior). The model learns the patterns and relationships

between the predictors and the outcome to estimate the probability of belonging to a specific segment.

2. **Predicting Propensity**: Once the model is trained, it can be used to predict the propensity or probability for each customer to belong to a particular segment. The input to the model would be the customer's predictor variables, and the output would be the predicted probability of segment membership.
3. **Assigning Propensity Scores**: The predicted probabilities from the propensity model can be transformed into propensity scores. Propensity scores represent the numerical values that indicate the strength of a customer's propensity for a specific segment. One common method for assigning scores is to rescale the predicted probabilities to a desired range, such as converting them to a scale of 0 to 100.
4. **Customer Segmentation**: Once the propensity scores are assigned, customers can be segmented based on their scores. For example, customers with high propensity scores may be assigned to a "high-value" or "loyal" segment, while those with lower scores may be assigned to a "potential" or "at-risk" segment. The specific segmentation criteria and cutoffs will depend on the business objectives and the characteristics of the customer base.

Hands-ON

To build a clustering model for customer segmentation using data from Google Analytics, you will need to perform the following steps:

Step 1: Import Required Libraries

```python
import pandas as pd
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
```

Step 2: Load the Data

Load the data from Google Analytics into a Pandas DataFrame. Ensure that the data contains the relevant features for demographic, behavioral, and geographical segmentation.

Step 3: Data Preprocessing

Perform necessary preprocessing steps, such as handling missing values, encoding categorical variables, and normalizing numerical features. You may also need to select the specific features to include in the clustering model.

Step 4: Feature Scaling

Standardize the data by scaling the features using the StandardScaler from scikit-learn. This step is crucial to ensure that the features are on a similar scale, preventing any one feature from dominating the clustering process.

```python
scaler = StandardScaler()
scaled_data = scaler.fit_transform(data)
```

Step 5: Determine Optimal Number of Clusters

Use a technique such as the elbow method or silhouette score to determine the optimal number of clusters for your data. This step helps you decide how many segments to create.

```python
# Example using the elbow method
inertia = []
k_values = range(1, 10)
for k in k_values:
    kmeans = KMeans(n_clusters=k, random_state=0)
    kmeans.fit(scaled_data)
    inertia.append(kmeans.inertia_)
```

```
# Plot the elbow curve
import matplotlib.pyplot as plt
plt.plot(k_values, inertia)
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.title('Elbow Curve')
plt.show()
```

Step 6: Train the Clustering Model
Choose the appropriate clustering algorithm, such as K-means, and train the model with the desired number of clusters.

Python
```python
# Example using K-means clustering
k = 4  # The optimal number of clusters determined from Step 5
kmeans = KMeans(n_clusters=k, random_state=0)
kmeans.fit(scaled_data)
```

Step 7: Assign Cluster Labels

Assign cluster labels to each data point based on the trained clustering model.

Python
```
cluster_labels = kmeans.predict(scaled_data)
```

Step 8: Analyze and Interpret Results

Analyze the cluster assignments and interpret the results. Explore the characteristics and behaviors of customers within each cluster to gain insights for marketing strategies and decision-making.

Note: The code provided is a general guideline for building a clustering model using the K-means algorithm. You may need to adapt it to your specific dataset and requirements, including any additional preprocessing or visualization steps.

To build a clustering model using demographic and behavioral features, such as age, gender, and number of transactions, you can follow these steps:

1. Data Preprocessing:
    - Load the dataset and perform an initial exploration to understand the structure and contents of the data.
    - Handle missing values by either imputing them or removing the corresponding records, depending on the extent of missingness and the impact on the analysis.
    - Convert categorical variables, such as gender, region, and device category, into numerical representation using techniques like one-hot encoding or label encoding.
    - Normalize the numerical variables, such as age, users, and transactions, to bring them to a similar scale.
2. Feature Selection:
    - Analyze the relevance and importance of each feature for clustering. This can be done through various techniques such as correlation analysis, feature importance from machine learning models, or domain knowledge.
    - Select the most relevant features based on the analysis and discard any irrelevant or redundant features. This step helps in reducing the dimensionality of the dataset and improving the clustering performance.
3. Hyperparameter Tuning:

- Choose a clustering algorithm suitable for your data, such as k-means, hierarchical clustering, or DBSCAN.
- Set the hyperparameters of the chosen algorithm, such as the number of clusters for k-means or the distance threshold for DBSCAN.
- Perform hyperparameter tuning using techniques like grid search or random search, optimizing a chosen evaluation metric (e.g., silhouette score or within-cluster sum of squares).

4. Clustering Model Training and Evaluation:
    - Split the preprocessed data into training and testing sets, ensuring that the testing set is not used during any training or hyperparameter tuning steps.
    - Fit the clustering algorithm to the training data and obtain the cluster assignments for the testing data.
    - Evaluate the clustering performance using appropriate metrics, such as silhouette score or within-cluster sum of squares, to assess the quality of the obtained clusters.
    - Analyze and interpret the resulting clusters based on the demographic and behavioral characteristics of the customers.

Note: The "Date" field is not typically used directly for clustering, as it represents time and does not provide meaningful information for grouping customers. If there are other relevant fields in your dataset that could be used for clustering, please provide them, and I can incorporate them into the model-building process.

Check this [python]