NAAN MUDHALVAN

APPLIED DATA SCIENCE

CUSTOMER SEGMENTATION USING DATASCIENCE (PHASE -2)

WHAT IS CUSTOMER SEGMENTATION?

In marketing, *customer segmentation* is the process of grouping customers by common traits. Discerning buying habits by customer type helps to market appropriately. For instance, it reveals the sizes of the various segments, how much we make from them, etc. This can help decide how to apportion the marketing budget.

In data science, *clustering* is the process of grouping objects by some common traits.

Single Structured Factor

First consider a single trait, say *industry*. We wish to segment customers just by this trait. First, let's assume *industry* has a small number of clean values. ("Clean" means from a controlled vocabulary — such as ones in a drop-down list.)

This segmentation is easy to do. It just involves grouping and aggregating.

DATASETS:

Dataset: www.kaggle.com

Dataset name: CUSTOMER SEGMENTATION USING DATA SCIENCE

Dataset link: https://www.kaggle.com/datasets/akram24/mall-customers

DETAILS ABOUT THE COLUMNS:

It consist of 4 columns and every column consist of 200 data's

1.Gender 2.Age 3. Annual income 4.Spending (1-100)

LIBRARIES:

PACKAGE INSTALLATION (Note: All the packages are installed through command prompt)

Pandas is a powerful library for data manipulation and analysis. It provides data structures like DataFrames for working with structured data, such as tabular data in CSV or Excel files.

INSTALLATION- pip install pandas

NumPy

Numpy is used for numerical computations and working with arrays.

INSTALLATION- pip install numpy

matplotlib

Matplotlib is a popular library for creating static, animated, or interactive visualizations in Python. The pyplot module provides a simple interface for creating various types of plots and charts.

INSTALLATION- pip install matplotlib

seaborn

Seaborn is built on top of matplotlib and provides a high-level interface for creating attractive and informative statistical graphics.

INSTALLATION- pip install seaborn

This line imports the `train_test_split` function from the `model_selection` module of the scikit-learn library. It's used to split the dataset into training and testing sets.

INSTALLATION- pip install scikit-learn

HOW TO TEST AND TRAIN:

Scikit-learn alias **sklearn** is the most useful and robust library for machine learning in Python. The **scikit-learn library** provides us with the model_selection module in which we have the splitter function train_test_split().

Syntax - from sklearn.model_selection import train_test_split

Load your dataset and prepare it for splitting. This typically involves loading the data, preprocessing it (e.g., handling missing values, encoding categorical variables), and separating the features (X) and the target variable (y). Use the **train_test_split** function to split your data into training and testing sets. Specify the proportion of data you want to allocate to the training set using the **test_size(test_size=0.2)**

80% training set

20% testing set

Train Your Model:

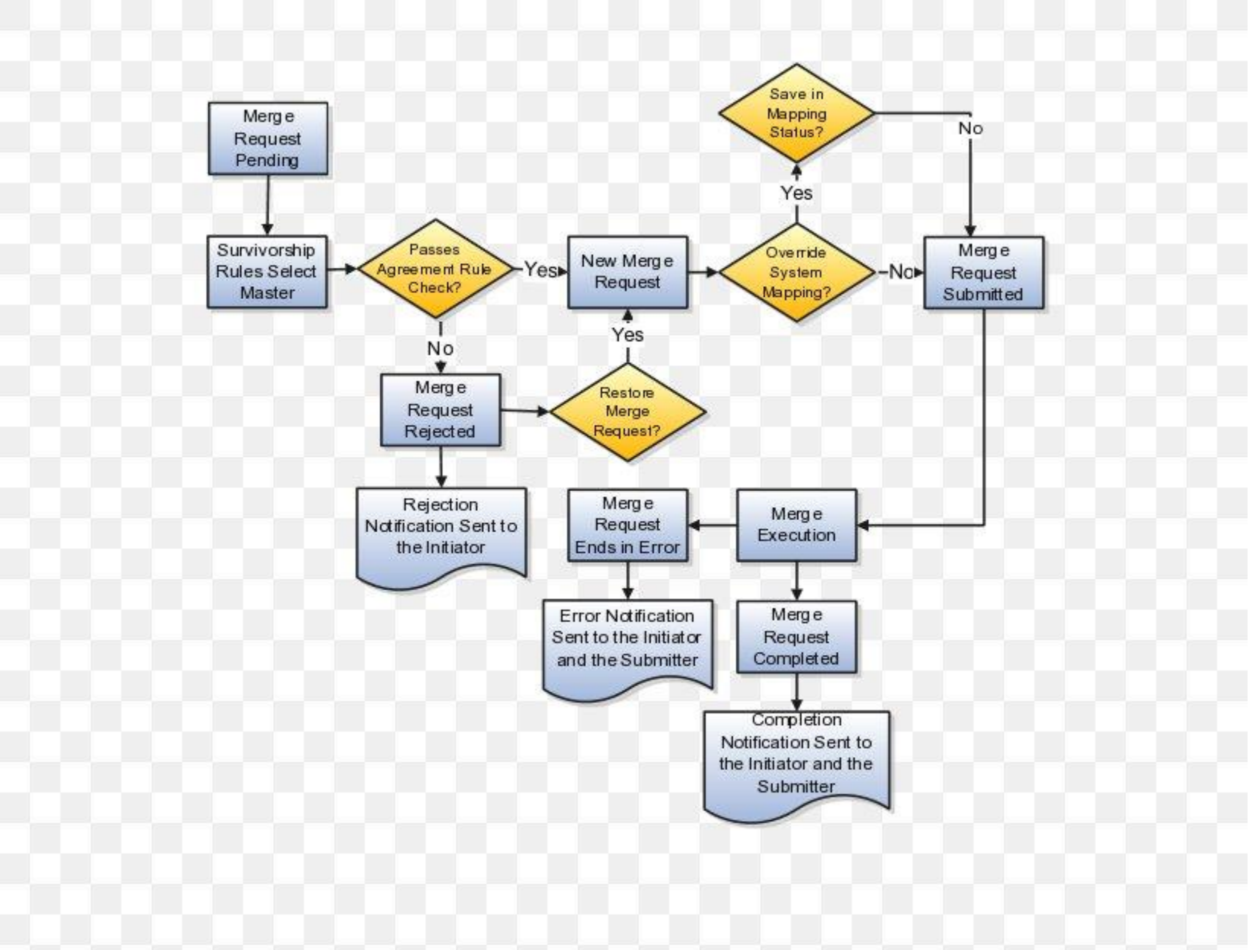
Use the training data (**X_train** and **y_train**) to train your machine learning model.

```
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

Evaluate Your Model:

```
y_pred = model.predict(X_test)
```

PROCESS FLOW:


Merge Request Pending
Survivorship Rules Select Master
Passes Agreement Rule Check?
Yes
No
New Merge Request
Override System Mapping?
Yes
Save in Mapping Status?
No
No
Merge Request Submitted
Merge Request Rejected
Restore Merge Request?
Yes
Rejection Notification Sent to the Initiator
Merge Request Ends in Error
Merge Execution
Error Notification Sent to the Initiator and the Submitter
Merge Request Completed
Completion Notification Sent to the Initiator and the Submitter

LINEAR REGRESSION:

Linear regression is one of the easiest and most popular Machine Learning algorithms. It is a statistical method that is used for predictive analysis. Linear regression is a supervised learning algorithm that compares input (X) and output (Y) variables based on labeled data. It's used for finding the relationship between the two variables and predicting future results based on past relationships.