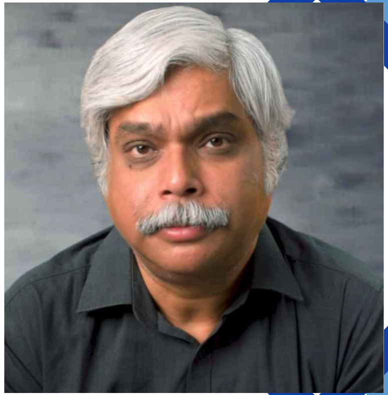
Clustering

Meet Your Speaker



Dr. Abhinanda Sarkar
Academic Director at Great Learning

- Alumnus - Indian Statistical Institute, Stanford University
- Faculty - MIT, Indian Institute of Management, Indian Institute of Science
- Experienced in applying probabilistic models, statistical analysis, and machine learning to diverse areas
- Certified Master Black Belt in Lean Six Sigma and Design for Six Sigma in GE

Learning Objectives

By the end of this session, you should be able to:

- Recall the need for distance metrics to measure similarity between data points.
- Make use of K-means clustering to group data points that exhibit similar characteristics together.
- Compare different metrics to evaluate the quality of clusters obtained and interpret them via profiling.
- Identify patterns in high-dimensional data by reducing it to lower dimensions using t-SNE for ease of visualization.
- Apply K-means clustering to real-world problems to identify groups in the data for informed decision-making.

Agenda

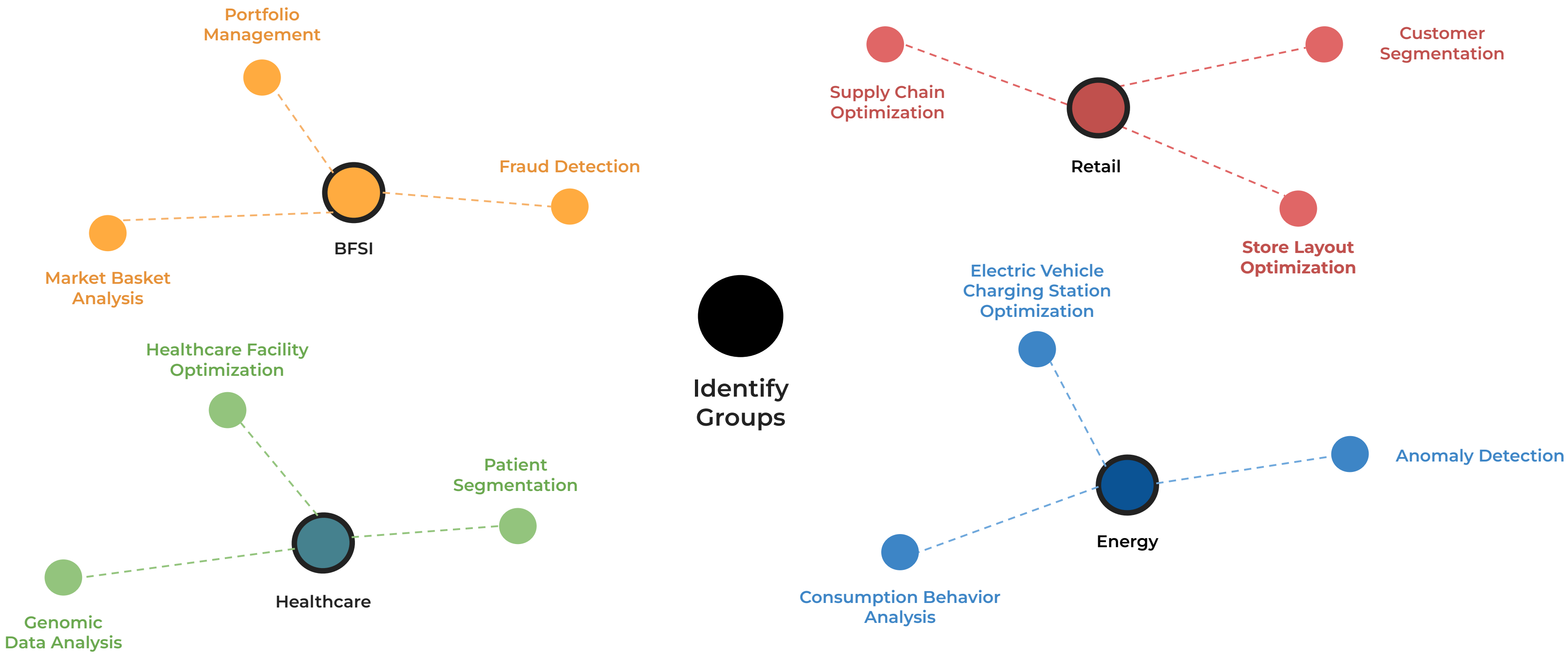
In this session, we'll discuss:

- Business Problems and Solution Space
- Distance Metrics
- Introduction to Clustering
- K-Means Clustering
- Optimal Number of Clusters and Cluster Profiling
- t-SNE for Visualization

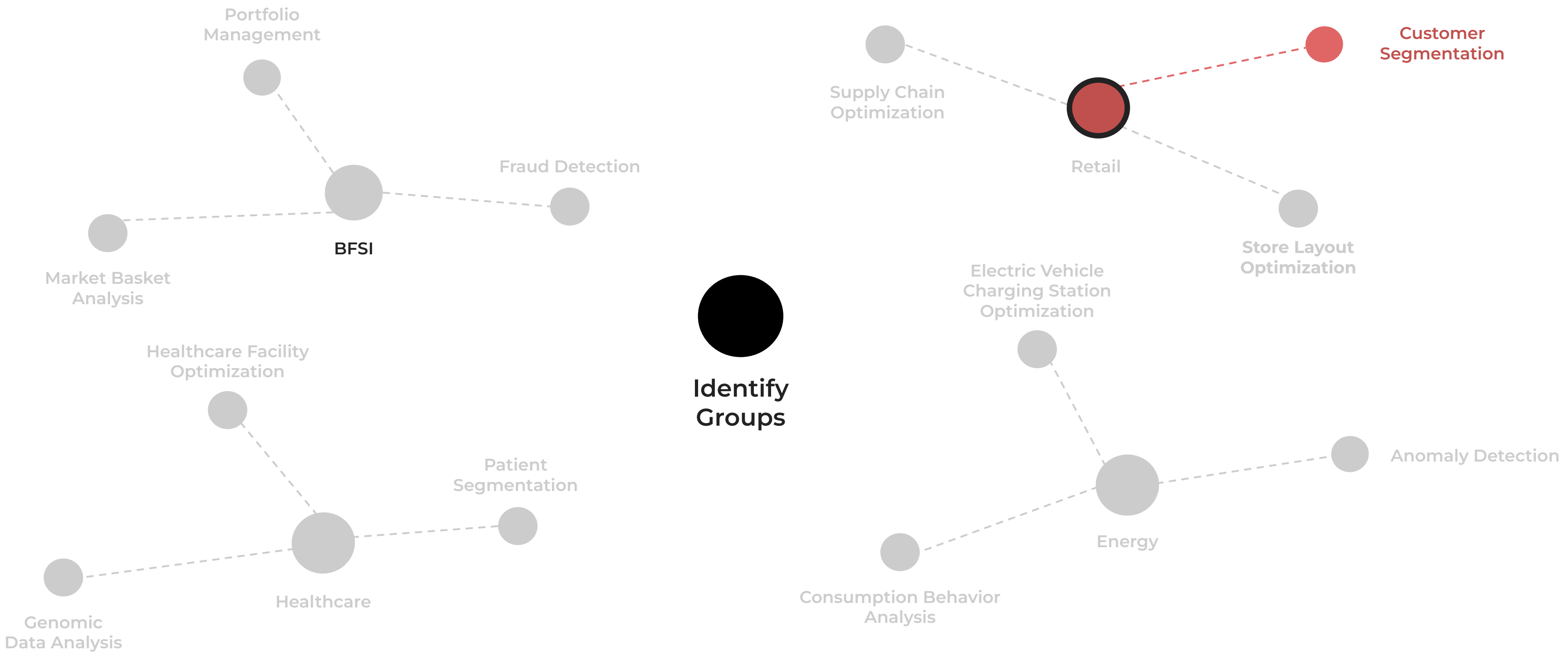
Common Business Questions

- How can we optimize the asset allocation for a portfolio by categorizing assets based on their risk and return profiles?
- How can we segment a customer base by their purchase and demographic attributes to develop targeted marketing campaigns?
- How can we group genetic profiles to identify patterns associated with specific diseases for medical research and drug development?
- How can we segregate geographical locations by energy consumption to detect unusual patterns and prevent power grid failures?

Problem Space



Problem Space



Problem Statement

- Consider a retail company aiming to gain a better understanding of their customer base using a customer segmentation strategy.
- Important to understand the characteristics of the customer segments for targeted marketing.
- Also helps to identify and develop retention strategies for high-value customers.

Objectives

- **Accurately segment customers based on their purchasing behavior**
- **Identify the key characteristics of each customer segment**

Customer Segmentation

Current State

Click-through Rate (CTR) for marketing campaign reduced by 4% as they do not cater to specific customer needs

3% increase in churn from old customers

Gap / Key Questions

What are the different segments of customers?

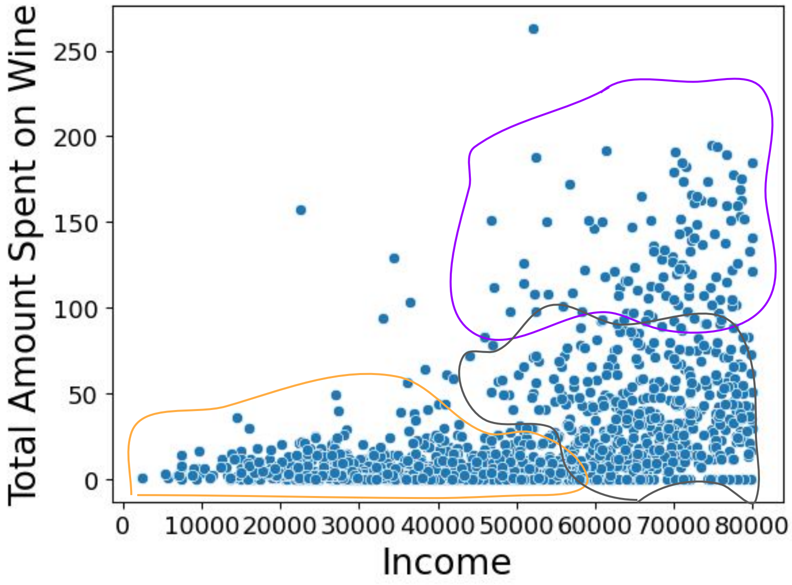
How do the different segments differ by purchasing behavior?

Desired State

20% increase in CTR due to targeted marketing guided by clear understanding of customer segments

Introduction of specialized offerings for loyal customers resulting in 10% less churn

Visualizing Relationships



- Income and Total Amount Spent on Wines are positively correlated.
- We can also observe some sort of ‘groups’ in the plot.

High income, High spending on wines

High income, Low spending on wines

Low income, Low spending on wines

Distance Metrics

- We observed groups in the data based on ‘closeness’ of points.
- Closeness (or distance) gave us a sense of similarity.

Points are ‘close’ => similar in nature

Points are ‘far’ => dissimilar in nature

How do we **quantify** this **closeness**?

Distance Metrics

- Need a **mathematical measure** to quantify closeness, i.e., distance.

A **distance metric** is a function, generally denoted by d(A,B), that **defines the distance** between the data points A and B as a **non-negative real number**.

d(A,A) = 0

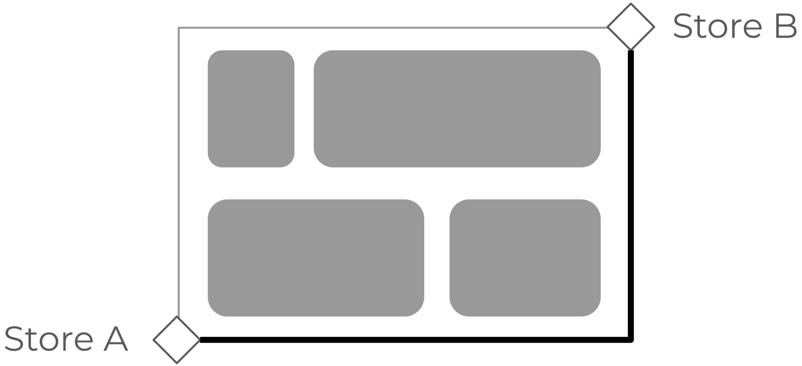
A is exactly similar to itself.

d(A,B) = d(B,A)

If A is similar to B, then B is similar to A.

Distance Metrics

- We want to travel from Store A to Store B.
- Grid lines represent roads; Space between grid lines are buildings.
- The path we'll take if we take a cab.

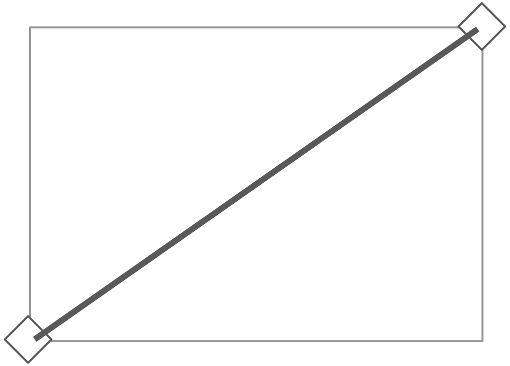


This is known as **Manhattan distance**.

$$d(x,y) = \sum_{i=1}^{n} |x_i - y_i|$$

Distance Metrics

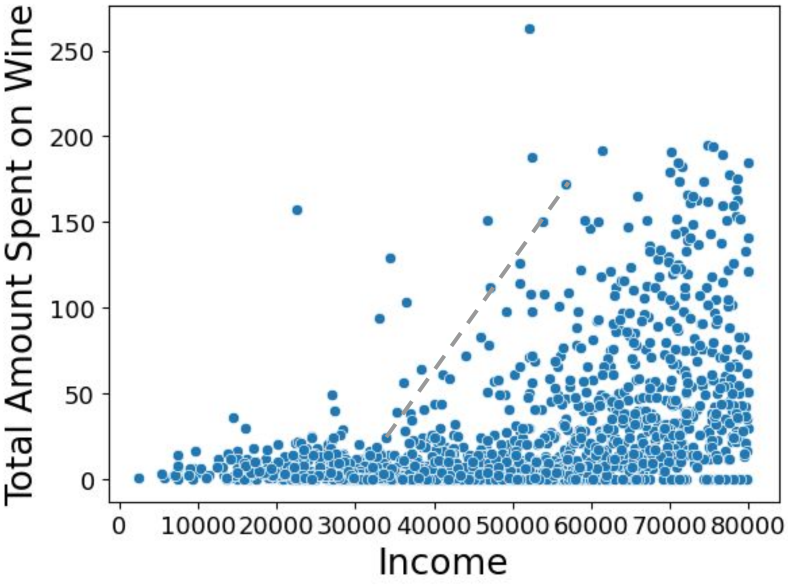
- Let's say we don't have buildings in between.
- Then we can take the **straight line path.**



This is known as **Euclidean distance**.

$$d(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

Distance Metrics



- Consider the points in the plot between which we want to find the distance.
- Note that the scale of the two attributes are very different.
- This will impact the distance computation.
- The attribute with a larger scale (Income) will dominate the computation.

How to solve this problem?

Distance Metrics

- We need to **scale the data** – bring all attributes to a similar range.
- For each attribute, compute the mean and standard deviation.
- For each value of a given attribute, **subtract the mean** and **divide by the standard deviation.**

$$Z = \frac{x - \mu}{\sigma}$$

This method is known as **Z-score scaling.**

Distance Metrics

Income	Total Amount Spent on Wine
58138.0	135
46344.0	11
71613.0	226
26646.0	11
58293.0	173

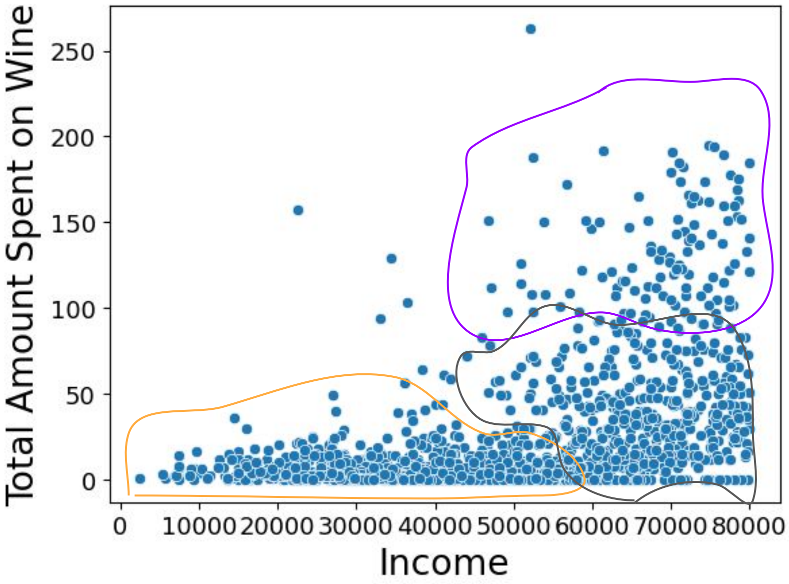
Original Data

Z-Score Scaling

Income	Total Amount Spent on Wine
0.506072	0.644281
-0.149978	-0.857380
1.255628	1.473853
-1.245692	-0.857380
0.514694	-0.337718

Scaled Data

Clustering



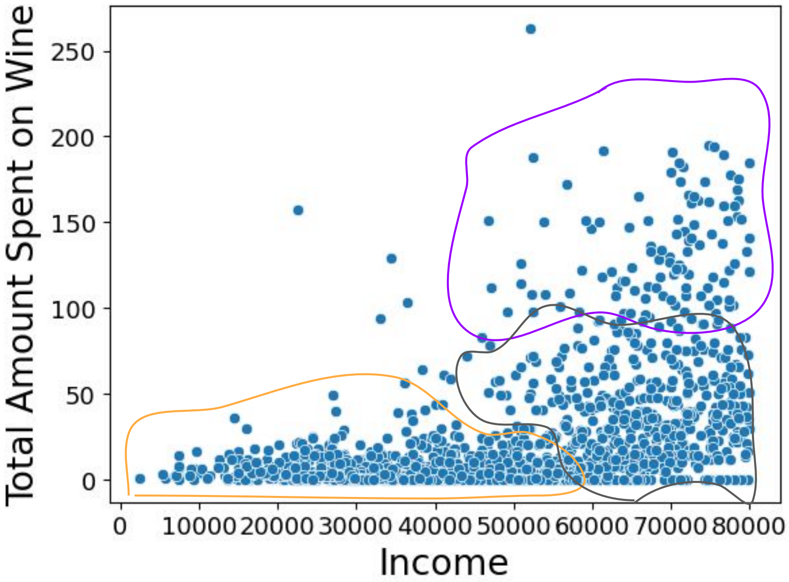
- We observed 'groups' in the data.

Points **in groups** are **similar**.

Points **across groups** are **dissimilar**.

- Need a **mathematical model** to do this.

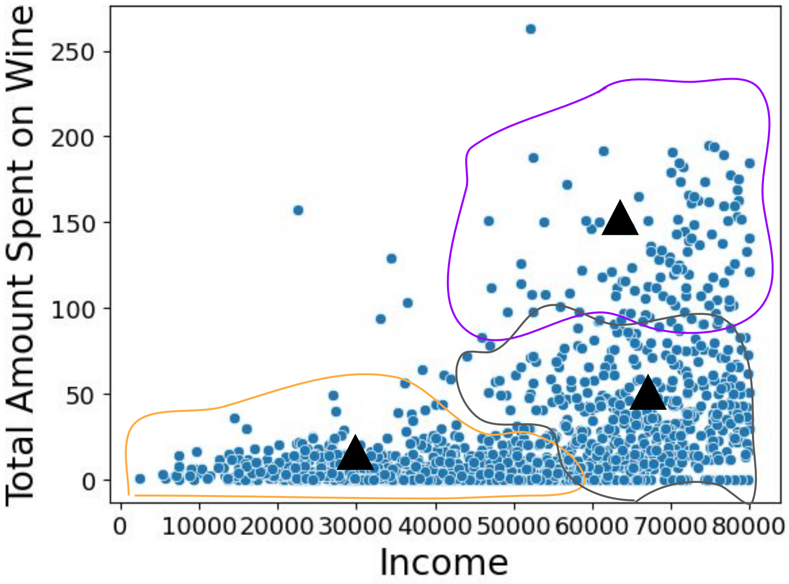
Clustering



- One approach could be to compute pairwise distances between all data points.
- Group pairs of points based on how close they are.
- Merge pairs to form bigger groups.
- Keep repeating till we reach a desired number of groups.

This is known as **connectivity-based clustering**.

Clustering



- Another approach could be to assume a certain number of groups.
- Often done based on visual analysis.
- We can define a ‘representative’ for each group.
- The points closest to these representatives can be grouped with them.

This is known as **centroid-based clustering**.

K-Means Clustering

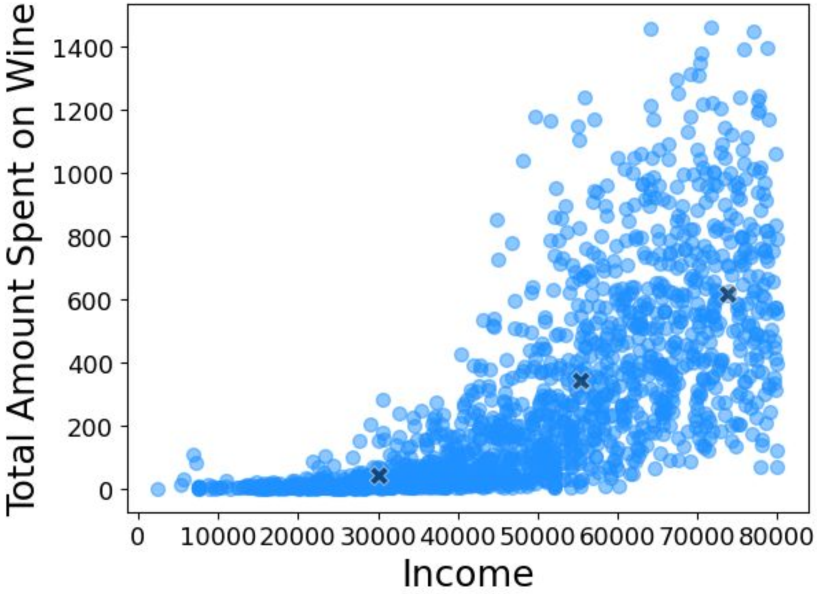
- How to choose the ‘representative’ of a group (or cluster)?
- A common way would be to choose the representative as a descriptive statistic that measure the ‘center’ of the data.
- The most common measure of ‘center’ of the data is mean.
- Assuming K clusters, we would want the representatives, or cluster centers, to be the mean of all the points in the cluster.

This method is known as **K-Means clustering**.

K-Means Clustering

1

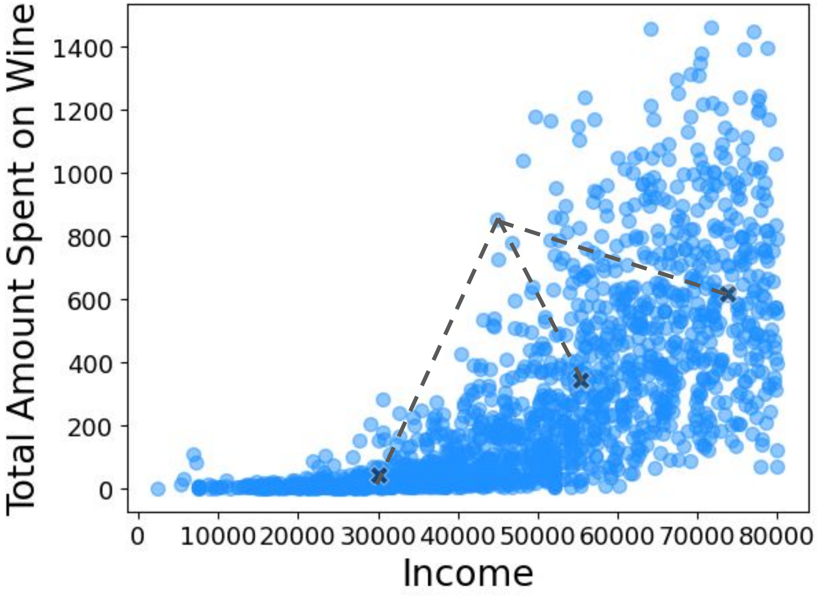
Randomly initialize K centroids.



K-Means Clustering

2

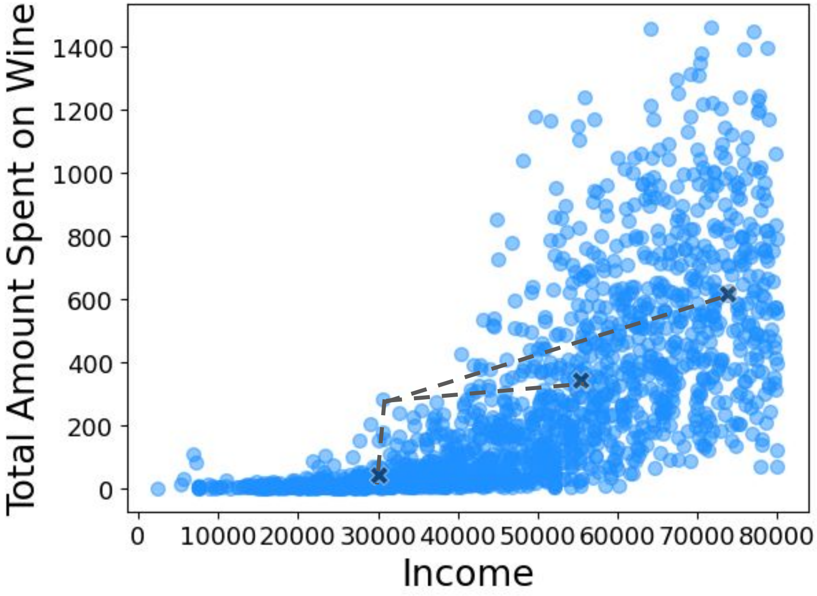
For each point, compute the distance to each centroid.



K-Means Clustering

2

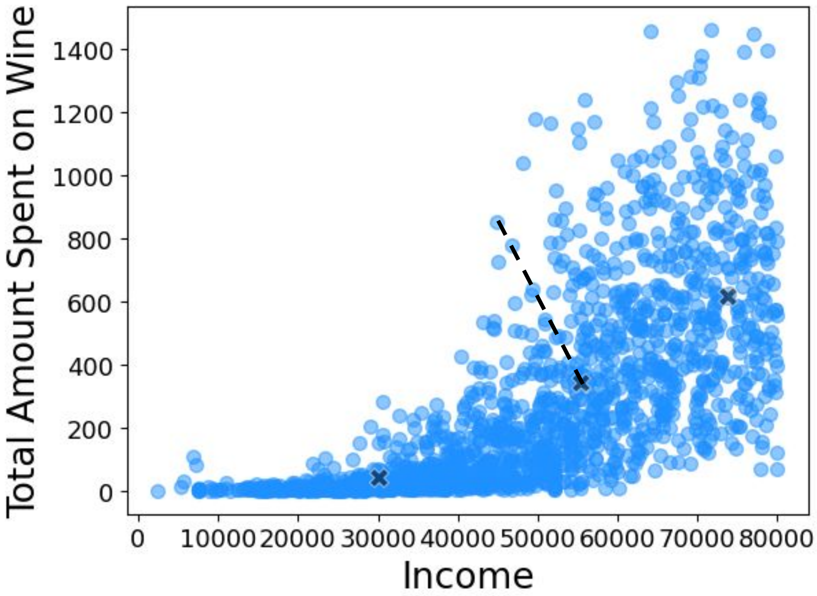
For each point, compute the distance to each centroid.



K-Means Clustering

3

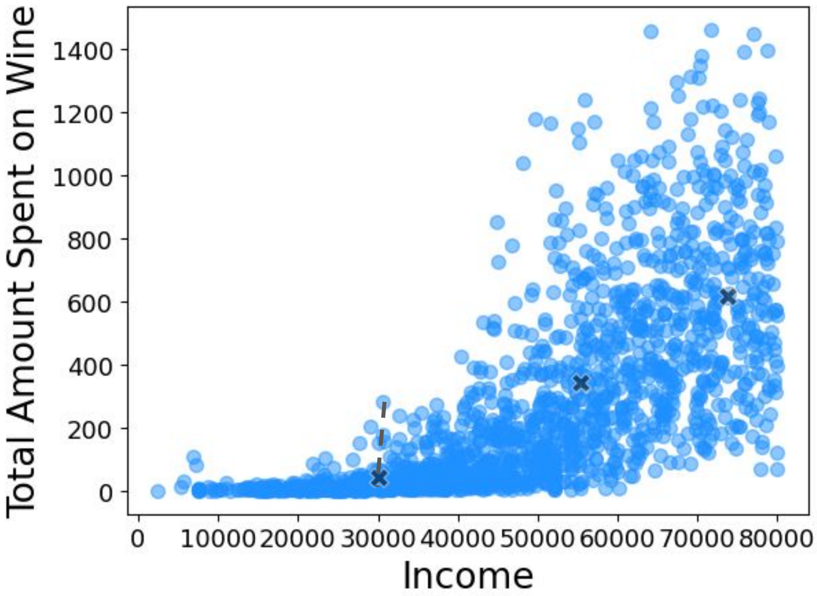
For each point, find the centroid with the minimum distance.



K-Means Clustering

3

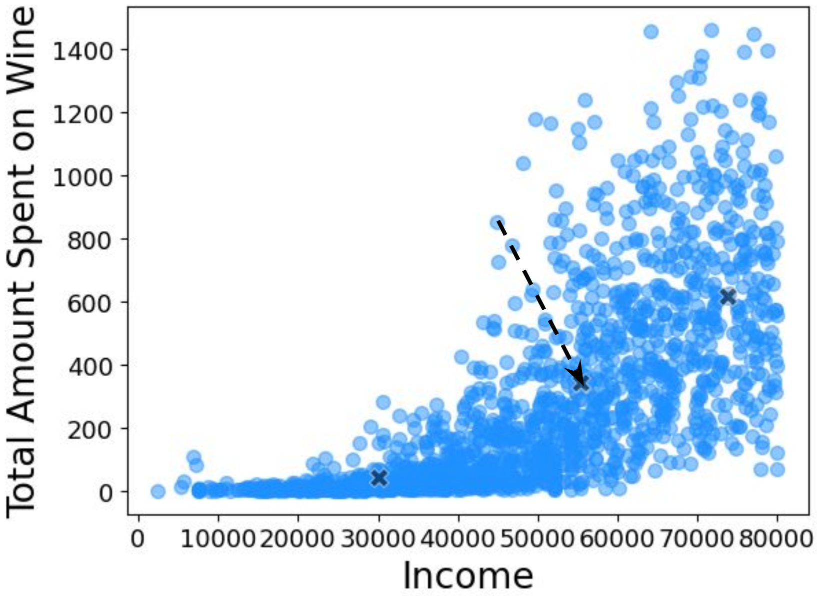
For each point, find the centroid with the minimum distance.



K-Means Clustering

4

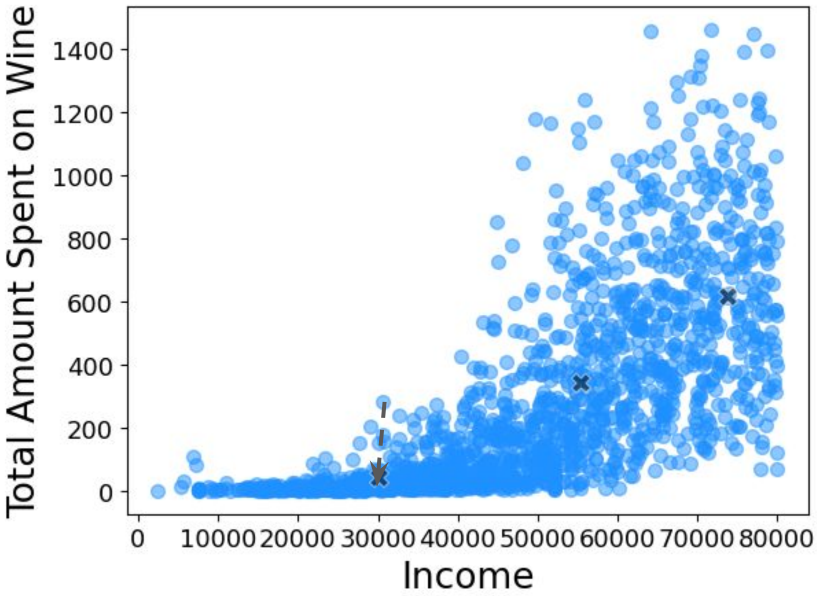
Assign each point to the nearest cluster centroid.



K-Means Clustering

4

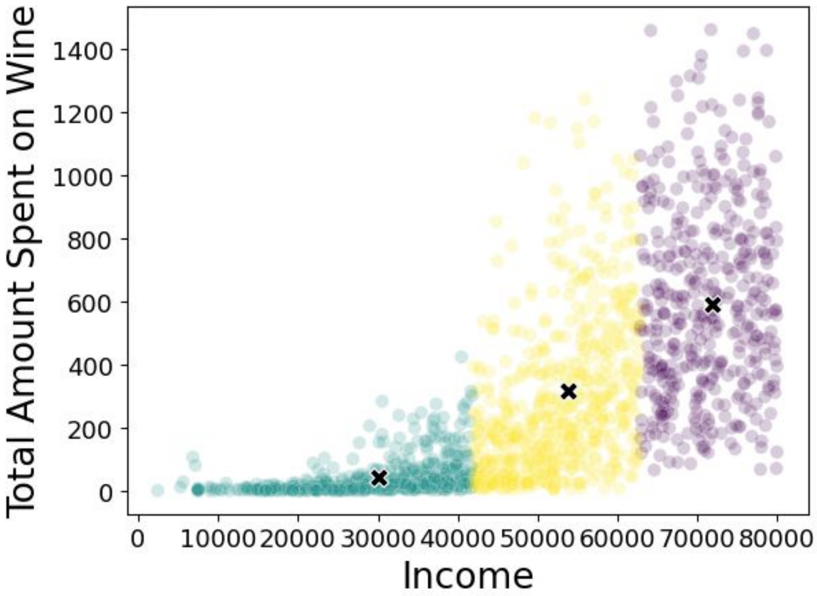
Assign each point to the nearest cluster centroid.



K-Means Clustering

4

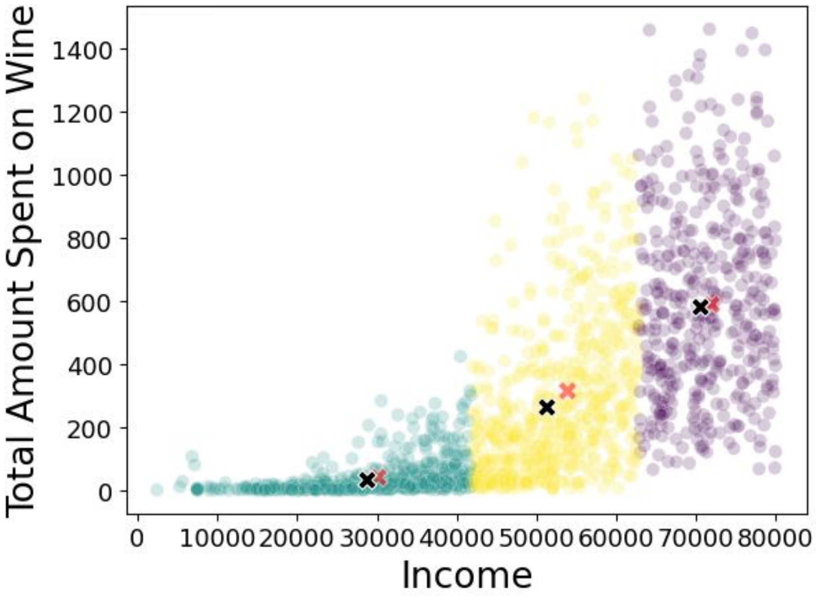
Assign each point to the nearest cluster centroid.



K-Means Clustering

5

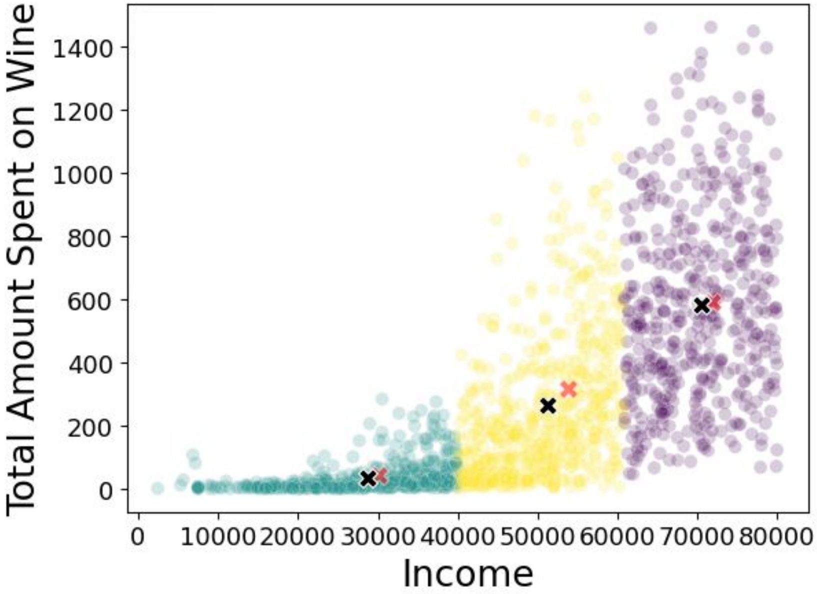
Recompute the centroids by taking the mean of the points assigned to each centroid.



K-Means Clustering

6

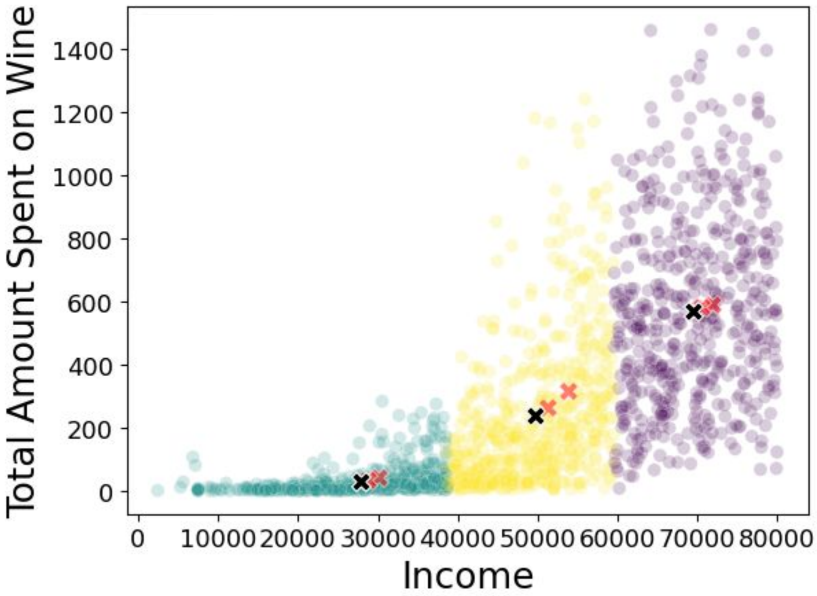
Repeat Steps 2 to 5.



K-Means Clustering

6

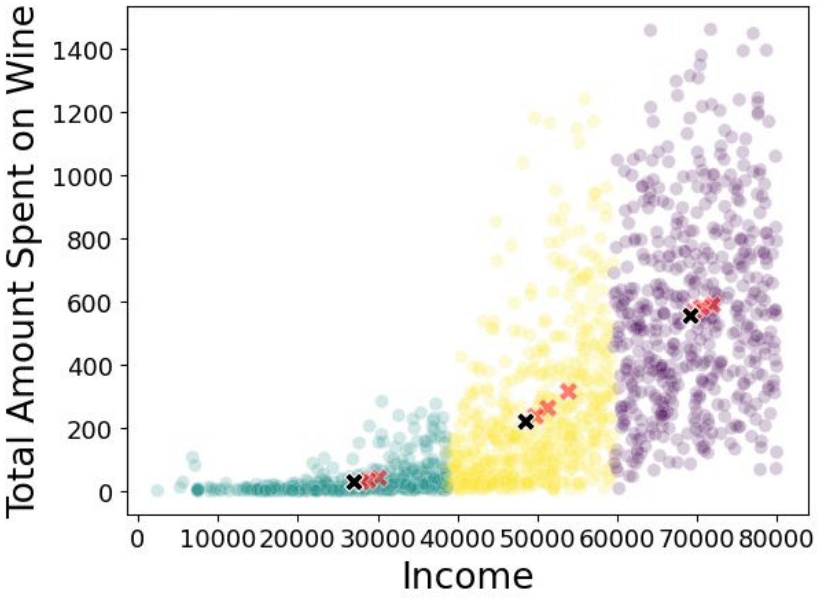
Repeat Steps 2 to 5.



K-Means Clustering

6

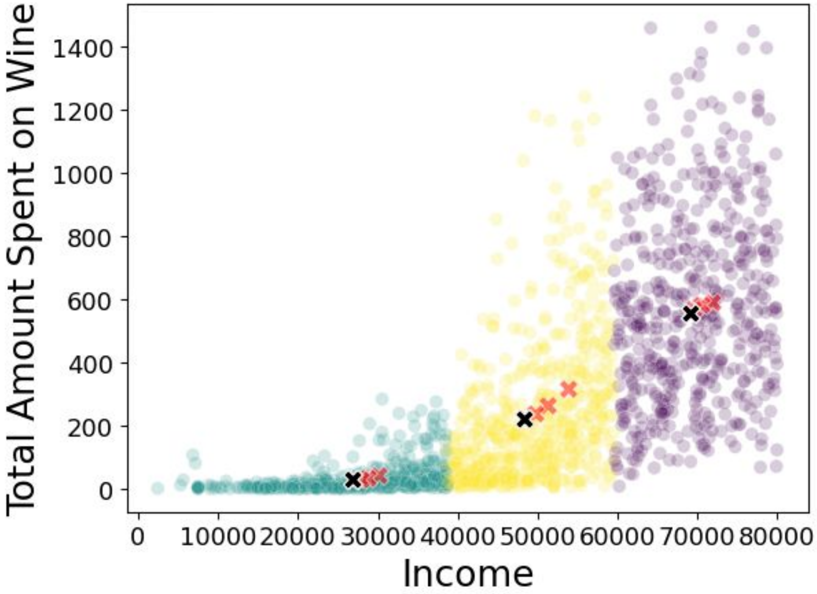
Repeat Steps 2 to 5.



When do we **stop**?

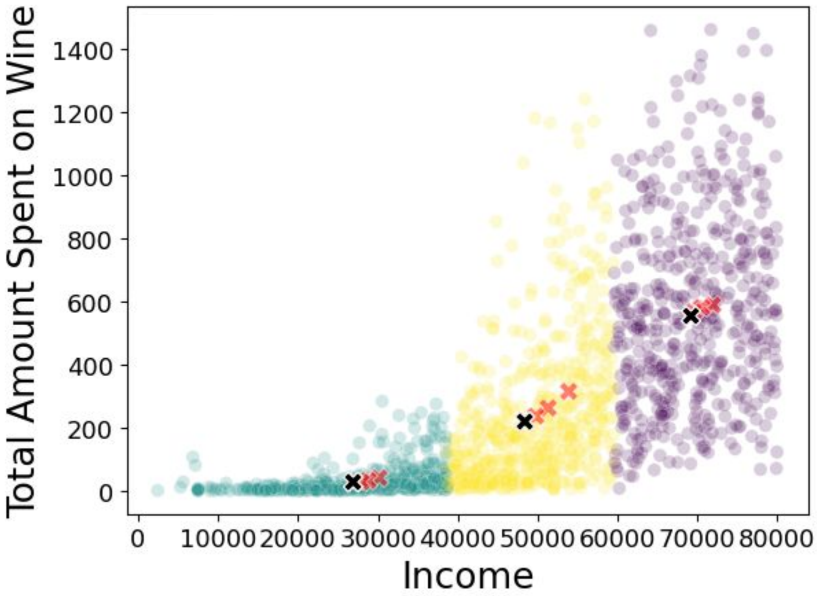
K-Means Clustering

- When the distance between the previous and current centroids is negligible.



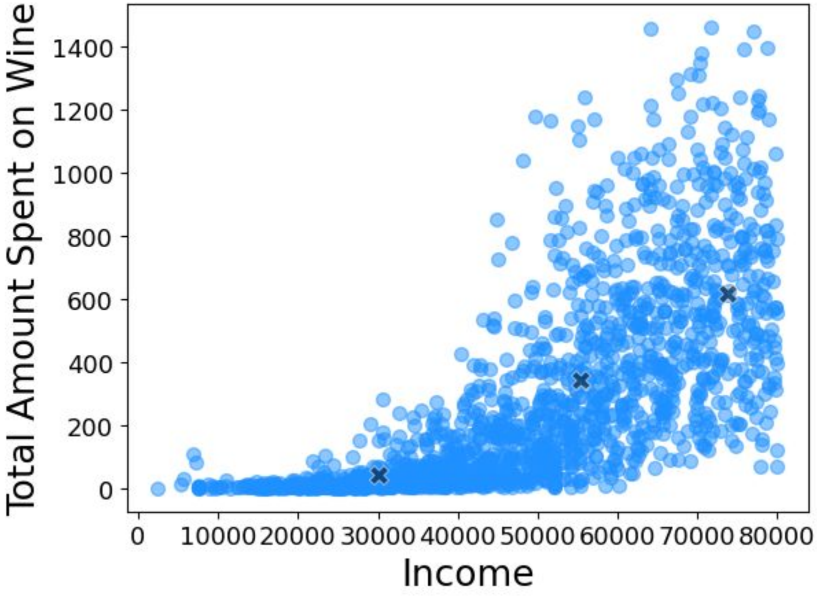
K-Means Clustering

- When the algorithm has run for a predefined number of iterations.



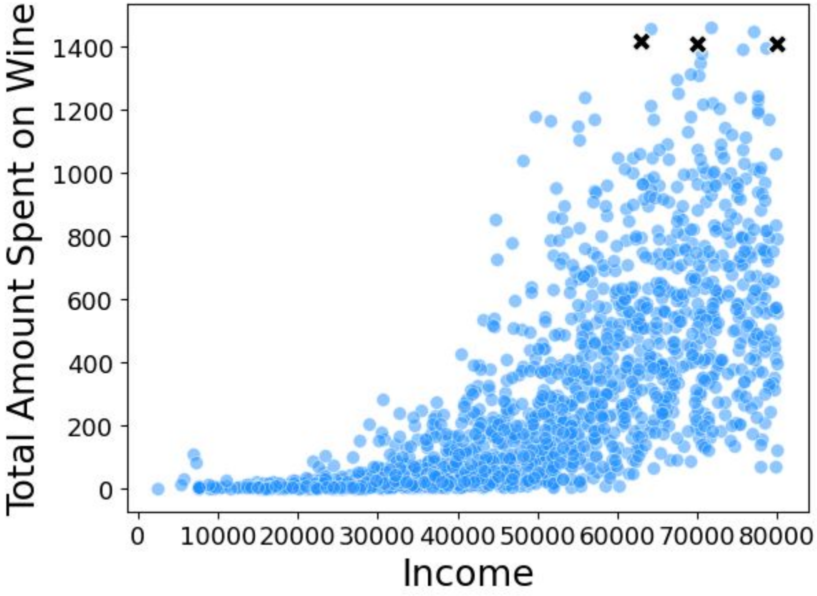
K-Means Clustering

- Need to initialize the centroids well at the start to get better clusters.



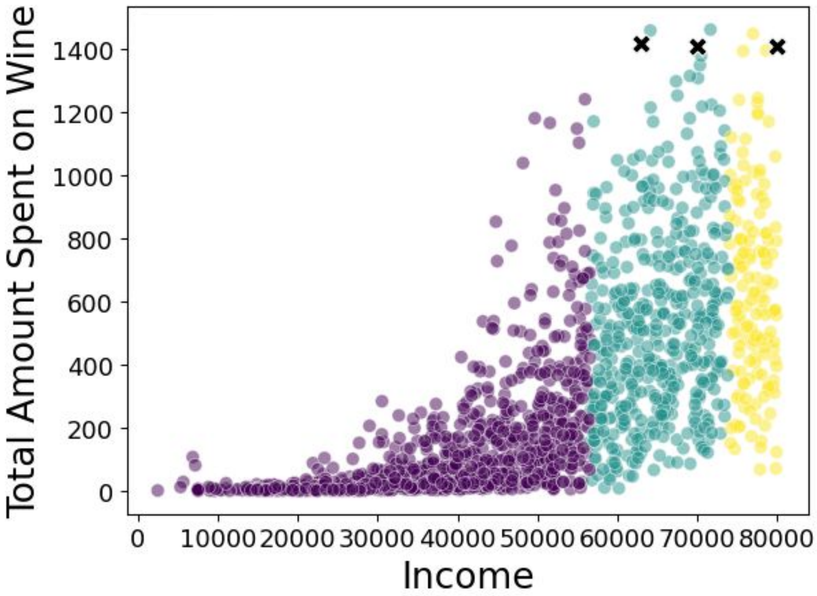
K-Means Clustering

- Need to initialize the centroids well at the start to get better clusters.
- Poor initialization can lead to suboptimal clusters.



K-Means Clustering

- Need to initialize the centroids well at the start to get better clusters.
- Poor initialization can lead to suboptimal clusters.



K-Means Clustering

Use a technique called **K-Means++** for **initialization.**

- The first centroid is randomly chosen from the data points.
- For each data point, compute its distance to the nearest centroid already chosen.
- The next centroids are selected such that the data points farther away from already chosen centroids are more likely to be selected as the next centroid.
- Repeat until all K centroids are.

Optimal Number of Clusters

- We start K-means clustering by assuming a certain number of clusters (K) in the data.
- How do we **determine** if we have chosen the **right K**?
- Recall the two goals we wanted to achieve with clustering.

Points **in groups** are **similar**.	Points **across groups** are **dissimilar**.

Optimal Number of Clusters

- One way would be to measure if points within a cluster are similar.
- For each point in a cluster, take the squared difference between the point and the cluster centroid.
- Penalize more for larger distances between a point and the cluster centroid.
- Take the sum of these squared differences for all points.
- As we have multiple clusters and each has a centroid, we can take the sum for all clusters.

This is called **Within-Cluster Sum of Squares** (**WCSS**).

Optimal Number of Clusters

- Provides a measure of the total variance within clusters.

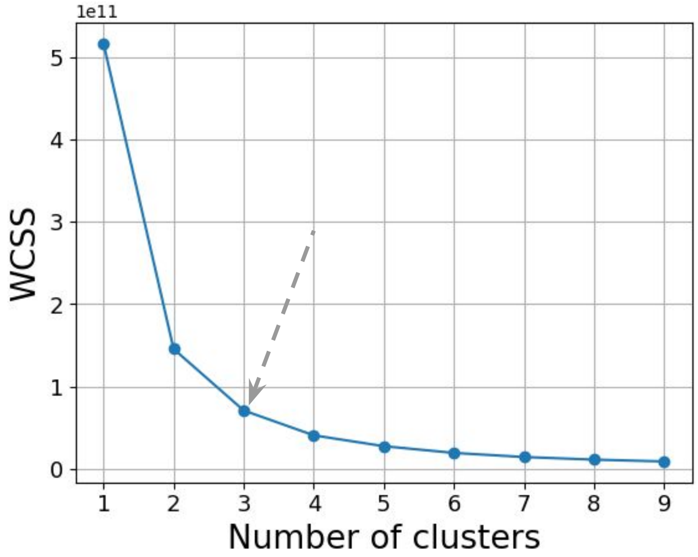
$$WCSS = \sum_{j=1}^{K} \sum_{x_i \in C_j} (x_i - \bar{x}_j)^2$$

- If we consider each point to be a cluster by itself, WCSS will be zero.
- If we consider all points to be a single cluster, WCSS will generally be high.

Optimal Number of Clusters

- We can compute WCSS for different number of clusters (K).
- Plot WCSS vs. K for the different values of K.
- We can observe a point where the rate of decrease of WCSS slows down sharply - like an ‘elbow’.
- Generally yields the optimal number of clusters for the data.

This is called the **Elbow method.**



Optimal Number of Clusters

- But WCSS captures only the variance within clusters.
- We also need to check if points in different clusters are dissimilar.
- Better to do both of these together instead of independently.

Optimal Number of Clusters

- Let's choose one cluster (say K) and a point say (i) within it.

Similarity within a cluster	Dissimilarity across clusters
Find the average distance from the point to all the other points in the cluster it belongs to.	Find the average distance between the point and all other points of the nearest cluster that the point is not a part of.
We denote this by a(i).	We denote this by b(i).

Optimal Number of Clusters

- We would want to minimize a(i) and maximize b(i) – low intra-cluster distance and high inter-cluster distance.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

This is called the **silhouette score.**

- Silhouette score of a cluster is taken as the mean of the silhouette scores of its data points.

Optimal Number of Clusters

- Silhouette score ranges between -1 and 1.

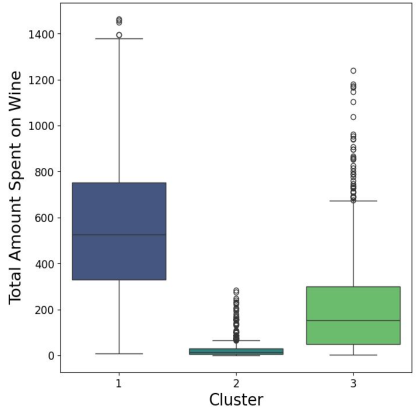
-1	0	+1
Poor clusters with potential wrong assignment of points	Overlapping clusters	Clearly separated clusters

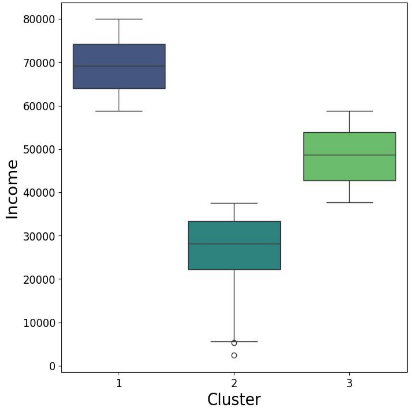
Cluster Profiling

- We have our optimal number of clusters
- Need to identify the characteristics of each cluster to get an understanding of the data within
- Use **cluster profiling**
- Helps analyze the clusters and identify their characteristics
- Helps us check if the clusters formed make business sense
- Can make informed business decisions based on these characteristics

Cluster Profiling

- We created 3 clusters from the data provided.
- Check boxplots of the attributes used for clustering.
- Segregate the boxplots by clusters.

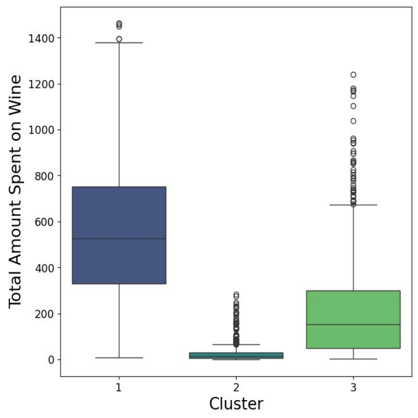


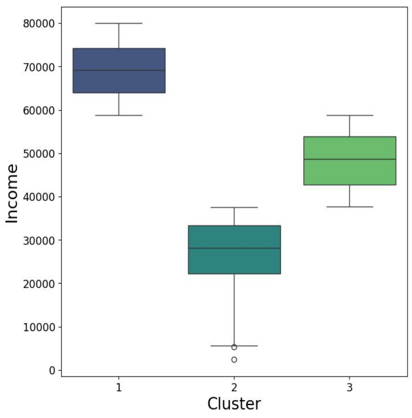


Cluster Profiling

Cluster 1

- Customers with high income
- Wide range in wine spending – some spend very less, some spend a lot

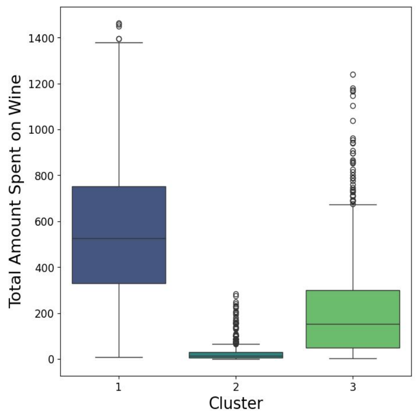


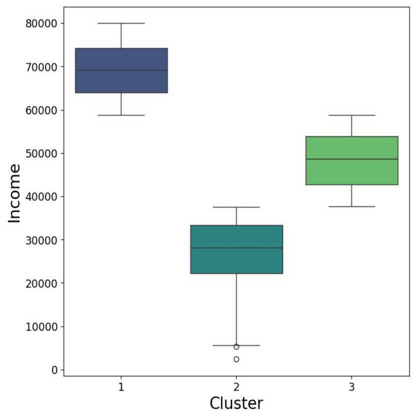


Cluster Profiling

Cluster 2

- Customers with low income
- Very low spending on wine

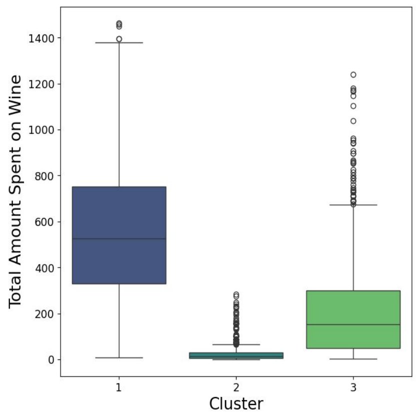


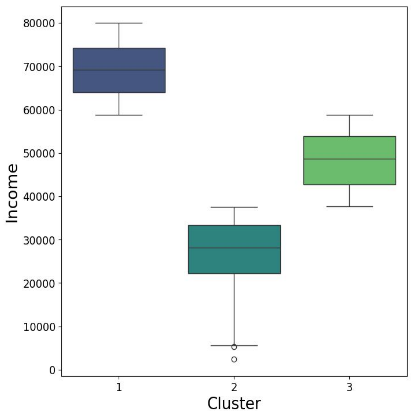


Cluster Profiling

Cluster 3

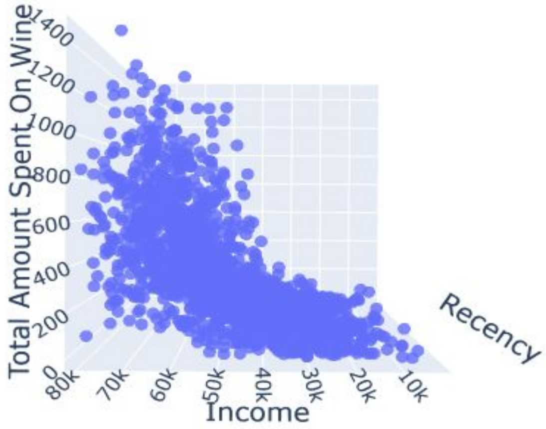
- Customers with medium income
- Wide range in wine spending, but not as much as Cluster 1 customers





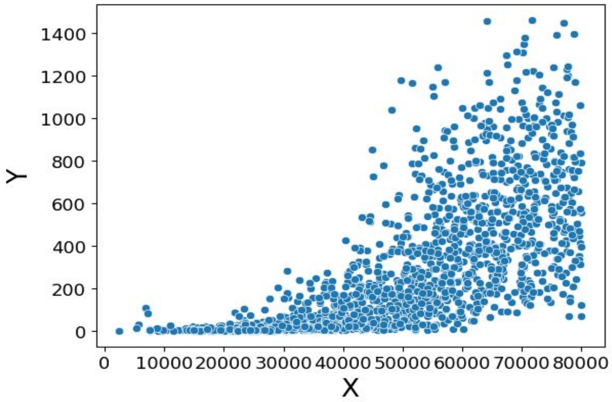
Need for Dimensionality Reduction

- Consider another attribute – Recency.
- We can visualize the data in a 3D plot to visually identify clusters.
- What if we have more than 3 attributes?
- **Can't visualize** data with **more than three dimensions.**



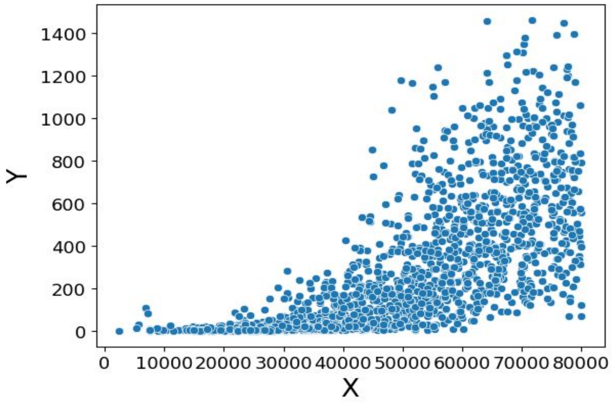
Need for Dimensionality Reduction

- One way to visualize high-dimensional data is to project it to a lower (2 or 3) dimension.
- Let's project our data from 3D to 2D.



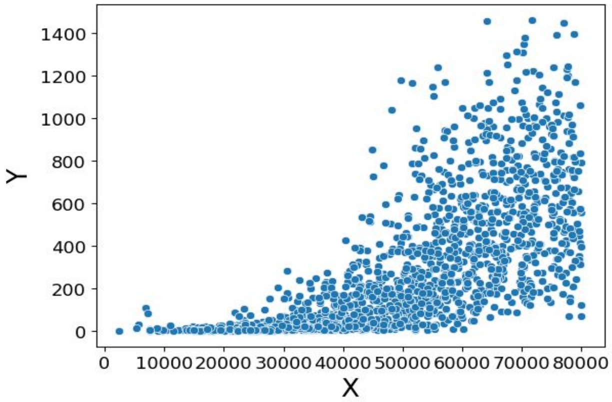
Need for Dimensionality Reduction

- The overall structure of the data seems to be preserved.
- But some regions that were dense in the higher dimension are sparse in the lower dimension.



Need for Dimensionality Reduction

- Some information was ‘lost’ when we moved from high to low dimension.
- In general, some information is lost when moving from high to low dimensions.
- Need to ‘reduce’ the ‘loss’ of information.



Need for Dimensionality Reduction

- One way to do this would be to ensure the following:

Points that are closer to each other in the higher dimension are close in the lower dimension too.	Points far apart from each other in the higher dimension are far apart in the lower dimension too.

- This will preserve the distribution of the data found in higher dimension to the lower dimension to a large extent.
- This is what **t-SNE** does.

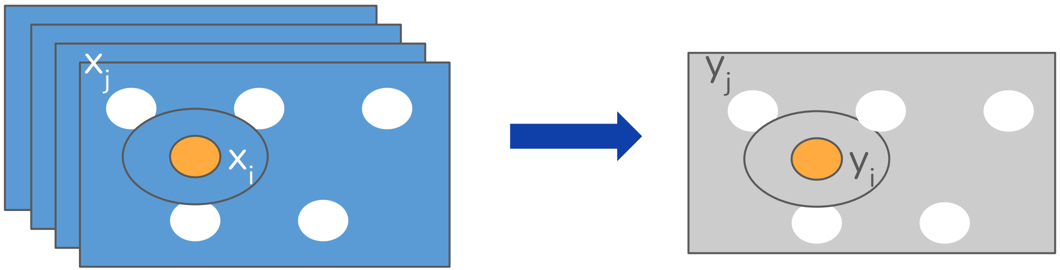
t-SNE

- **t-SNE** stands for **t**-distributed **S**tochastic **N**eighbor **E**mbedding
- A non-linear dimensionality reduction technique
- Can be used to map high-dimensional data to 2 or 3 dimensions
- Mainly used for visualization purposes

How does t-SNE work?

t-SNE

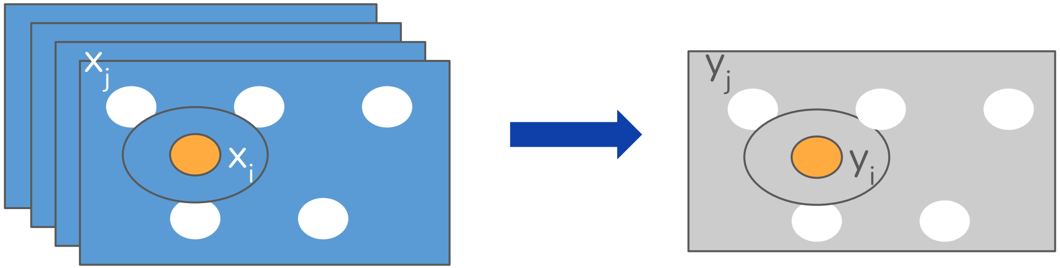
- Compute a distribution that measures pairwise similarities in original data (high dimension).
- Find a ‘close’ lower dimension mapping of pairwise similarities.
- Use this mapping to transform data from high dimension into low dimension.



t-SNE

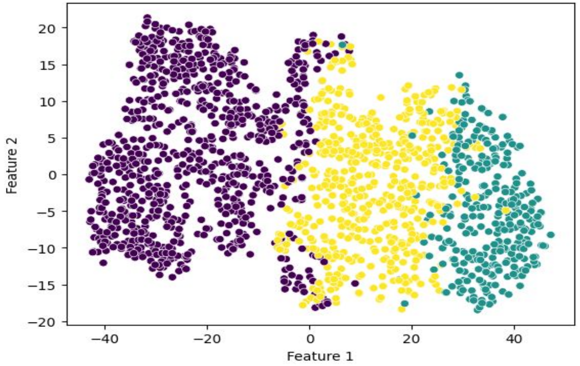
- Finding the 'closest' low dimension mapping involves minimizing the divergence between two distributions.

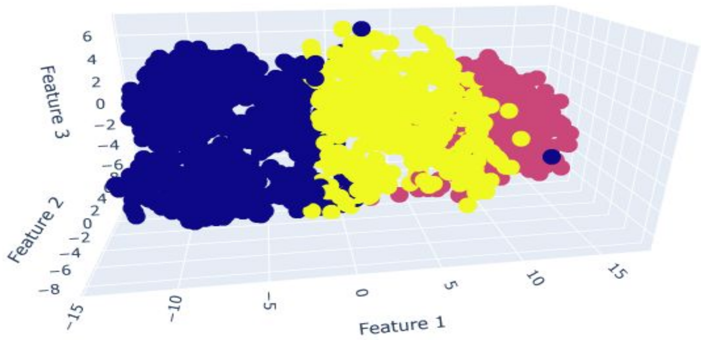
- Iteratively improve the lower-dimension mapping.



t-SNE

- Let's visualize the clusters obtained from multiple features after dimensionality reduction using t-SNE.

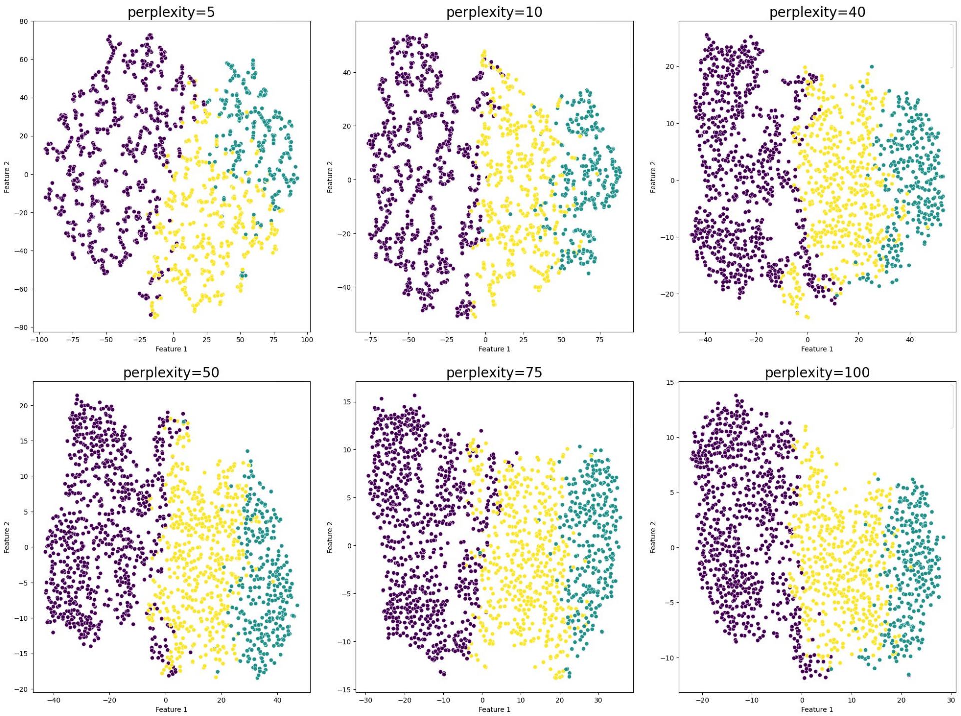




t-SNE

- Lower dimension features obtained are not interpretable.
- Can be used to visually identify clusters based on similarity of data points.
- The parameter **perplexity** can be fine-tuned to better represent the high-dimensional data in low dimensions.
- Provides a sense on the number of neighbors.
- Affects the quality of visualization.

t-SNE



- Let’s visualize the clusters using different perplexity values.

Summary

Here's a brief recap:

- Identify scenarios where clustering can be applied, and understand how clustering provides solutions by grouping similar data points together, facilitating better decision-making and insights.
- Distance metrics measure similarity or dissimilarity between data points in clustering.
- Clustering groups data points based on similarity in an unsupervised manner and ensure intra-cluster similarity and inter-cluster dissimilarity.
- K-Means Clustering divides data into clusters by minimizing within-cluster variance. It Initialize centroids, assign points, recompute centroids, repeat.

Summary

Here's a brief recap:

- Determining the optimal number of clusters is crucial. Methods such as the Elbow Method, Silhouette Score, and Gap Statistic are used to identify the right number of clusters (K).
- Cluster profiling involves analyzing and describing the characteristics of data groups formed by clustering algorithms to understand their distinct patterns.
- t-SNE visualizes high-dimensional data by reducing its dimensions. It reduces dimensions while preserving structure, making clusters visible.

Learning Outcomes

You should now be able to:

- Summarize the importance of distance metrics in measuring similarity between data points in clustering.
- Explain the rationale behind using K-means clustering to group data points with similar characteristics.
- Apply various distance metrics to assess similarity and dissimilarity between data points.
- Evaluate the quality of clusters obtained from K-means using metrics like silhouette score and interpret cluster profiles.
- Design clustering solutions using K-means tailored to specific real-world problems for data-driven decision-making.

Happy Learning !