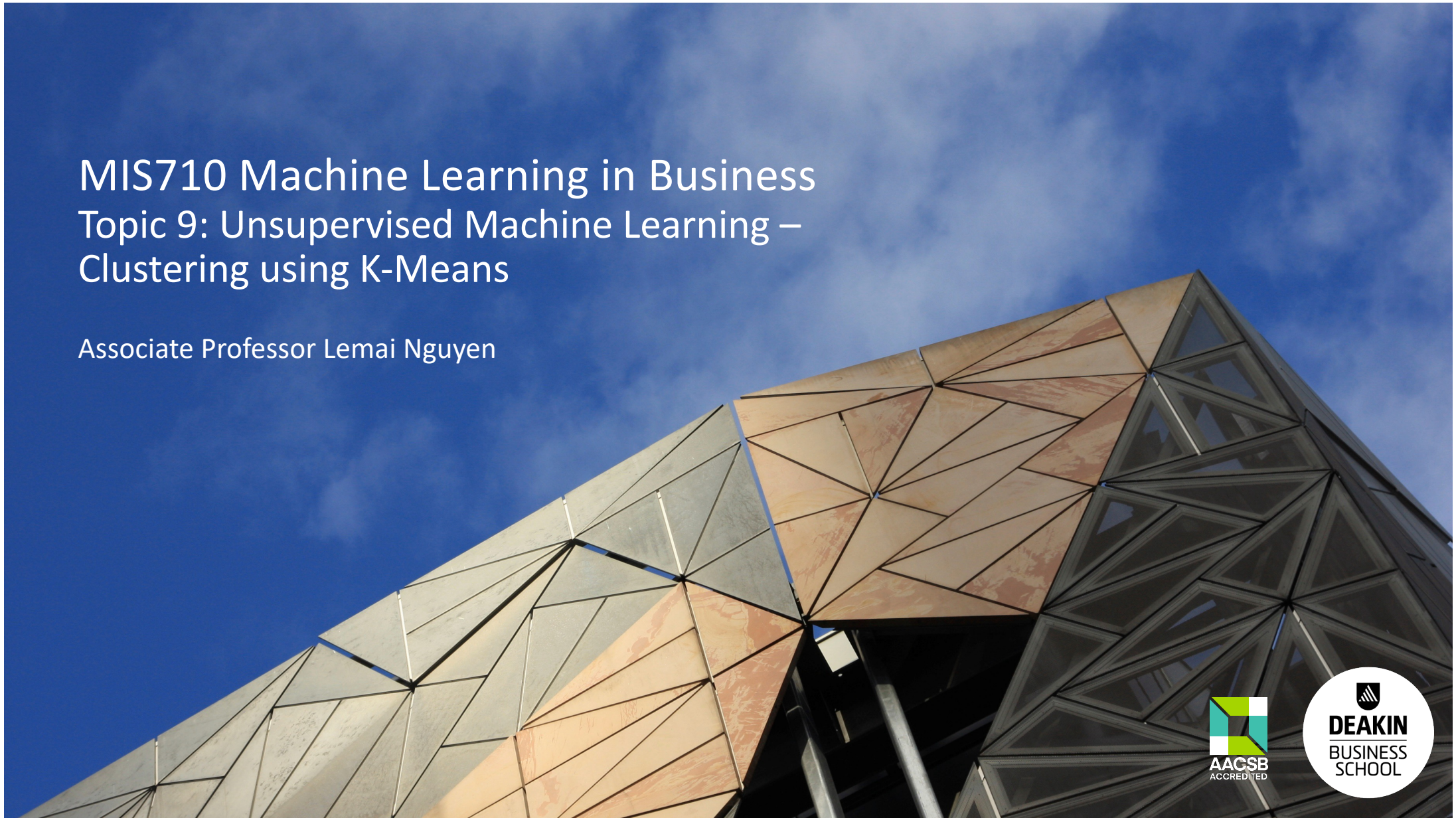


MIS710 Machine Learning in Business

Topic 9: Unsupervised Machine Learning – Clustering using K-Means

Associate Professor Lemai Nguyen

AACSB ACCREDITED

DEAKIN BUSINESS SCHOOL

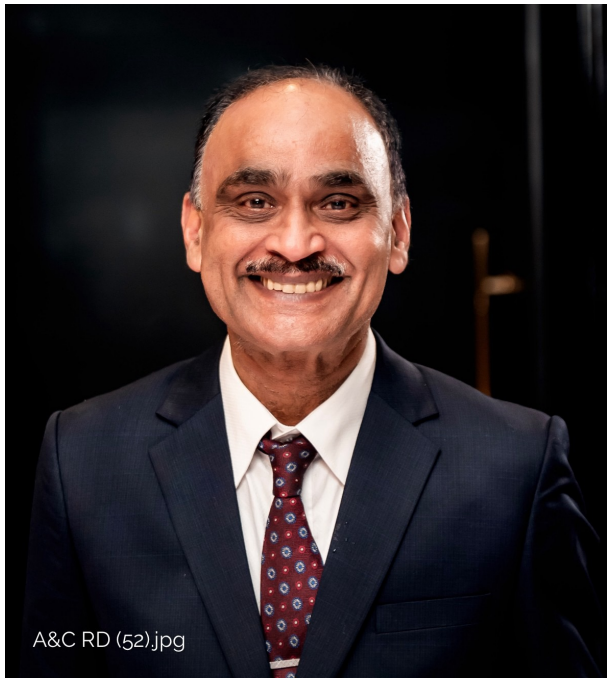
CONGRATULATION !!!

- Best A1 report: **Anna Mihaylov**
- First Prize: **Veena Suresh**
- Runner Up: **Rajvirsinh Jitendrasinh Rathod**
- Woman in AI:
  - **Chu Yin**
  - **Akshitha Karlapati**

and **ALL participants – Thank you!**

Dr. Raju Varanasi

Professor of Practice



Dr. Raju Varanasi commenced as Professor of Practice in the Department of Information systems and Business Analytics at the Deakin Business School. Dr. Varanasi brings a distinguished blend of educational excellence and industry expertise with a career that spans over three decades in Australia.

Dr. Varanasi has held pivotal roles in TAFE NSW, public and Catholic schools in leveraging digital technologies and data analytics for transformational impact. Dr. Varanasi is a Chemical Engineer an MBA from IIT, New Delhi and IIM Bangalore- world class universities in India. An Australian Fulbright Scholar, Dr. Varanasi attained his PhD from University of Newcastle with his thesis on transforming school systems. His visionary approaches to education have been recognized with awards such as the Top 50 Australian CIO Award (twice) and the Gartner Innovation in Education Award.

He is also a published author, contributing to the literature on educational analytics and school progress frameworks. After roles as a General Manager, Director, COO and CIO in the last 20 years, Dr. Varanasi is consulting for Google's partners across Asia Pacific advising on data, analytics & AI strategies. As a Professor of Practice, Dr. Varanasi is committed to forging stronger partnerships between academia and industry, preparing students to navigate and excel in the dynamic field of Analytics & AI to equip them with the skills necessary to lead enterprises in a data-driven future.

Dr. Raju Varanasi | LinkedIn

Learning Analytics for Primary Schools

A2

Case Study

A2 Tasks

A2 Deliveries

- **Six** EDA questions
- **Two** supervised ML models and evaluation comparison
- **One** clustering model and cluster profiling

- **Two** reports
  - Dr. Alok Sinha, Data2Intel Director of **Data and Insights** and team
  - Sally Tran, Data2Intel Director of Education and Engagement - **business** audience

- **Two** Python files
  - ipynb and PDF version of the Python notebook

Ethical recommendations – extended FAT framework

Learning Analytics for Primary Schools

A2

Case Study

A2 Tasks

A2 Deliveries

Fairness

- Sample size of a protected sub-population
- Does the protected sub-population have same distribution of predictions as others

Accountability

- Document how you select features, handle missing data, and choose model algorithms, and impacts on model performance
- Potential implications of the model, particularly to vulnerable sub-populations, suggest ethical and privacy guidelines

Transparency

- Interpret and explain how the model works and its performance metrics and trade off
- Ensure comprehensive documentation including data sources, preprocessing steps, model choices, and validation processes.

Privacy

- Data minimisation
- Access control

A2 consultation sessions

Learning Analytics for Primary Schools

A2

Case Study

A2 Tasks

A2 Deliveries

Week 10:

- Friday 20th Sep 3.00-4.00pm AEST i.e., 10.30-11.30am IST: Lemai - How to write a business report? Zoom

Week 11:

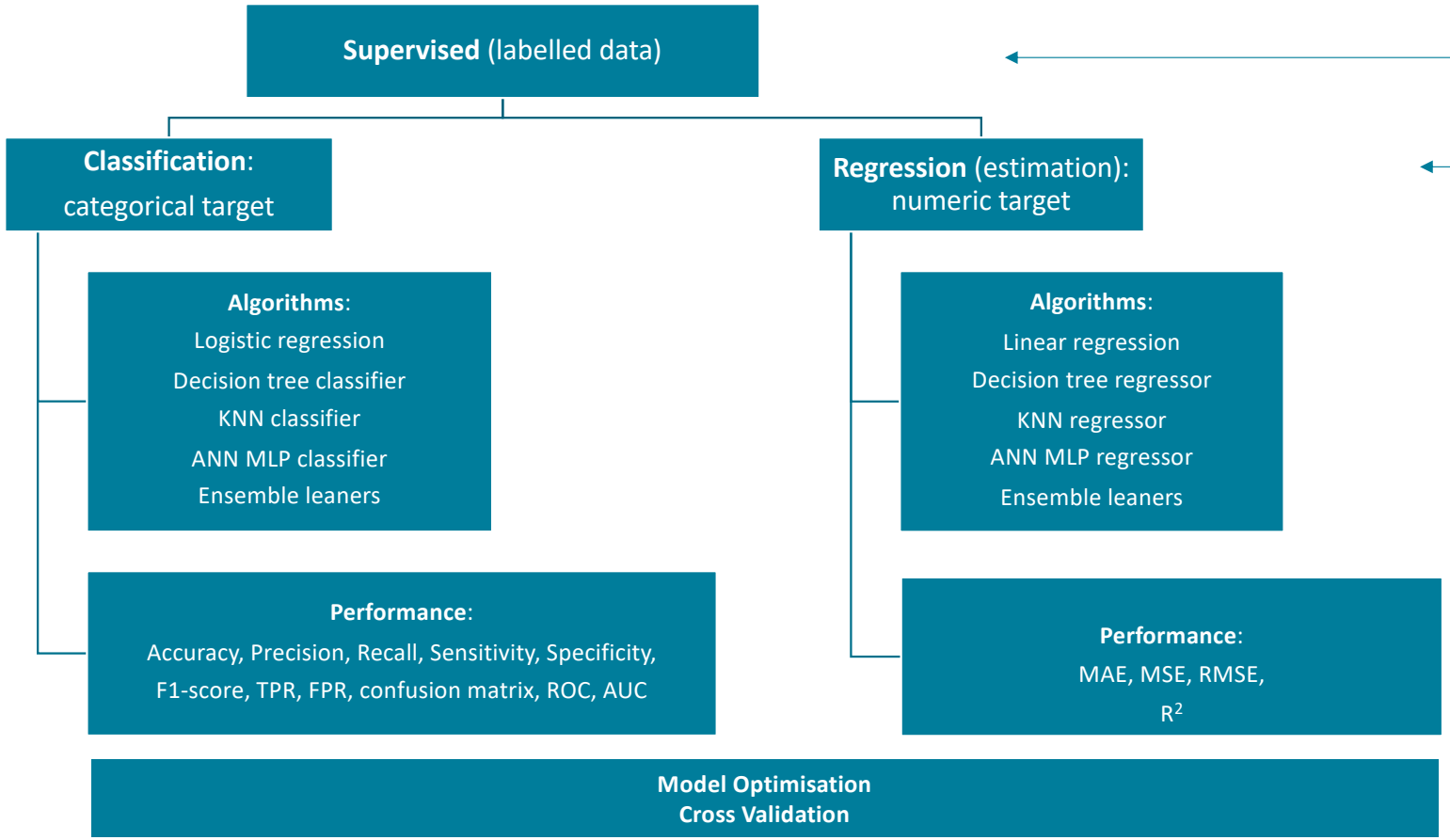
- Tuesday 24th Sep 3.00-4.00pm AEST, in person, Dat Le (venue TBA)
- Wednesday 25th Sep 7:30-8:30pm, 3.00-4.00pm IST, Zoom Emran
- Thursday 26th Sep 3.00-4.00pm AEST, in person, Thuc (venue TBA)

Week 12 - study week – Zoom only

- Monday 30th Sep 3.00-4.00pm AEST i.e., 10.30-11.30am IST, Zoom, Dat Le
- Tuesday 1st Oct 6.00-7.00pm i.e., 1.30-2.30pm IST, zoom, Durgesh
- Wednesday 2nd Oct 6.00-7.00pm i.e., 1.30-2.30pm IST, zoom, Emran

We discuss A2 in weekly classes, offer A2 consultations during weekly labs, and answer questions in the A2 forum!

Recap



Type of learning

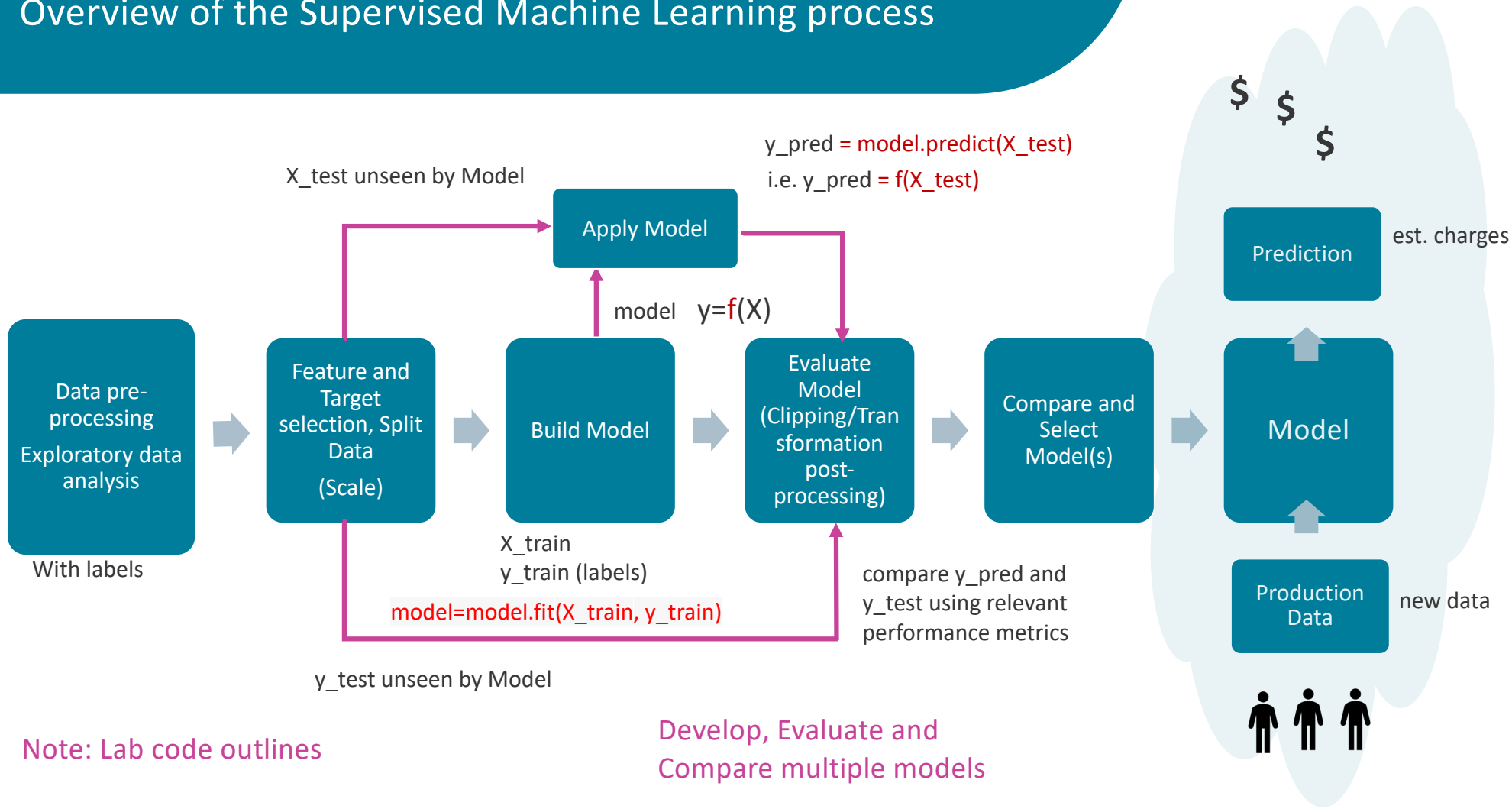
ML problems

ML algorithms (types of models)

Evaluation metrics

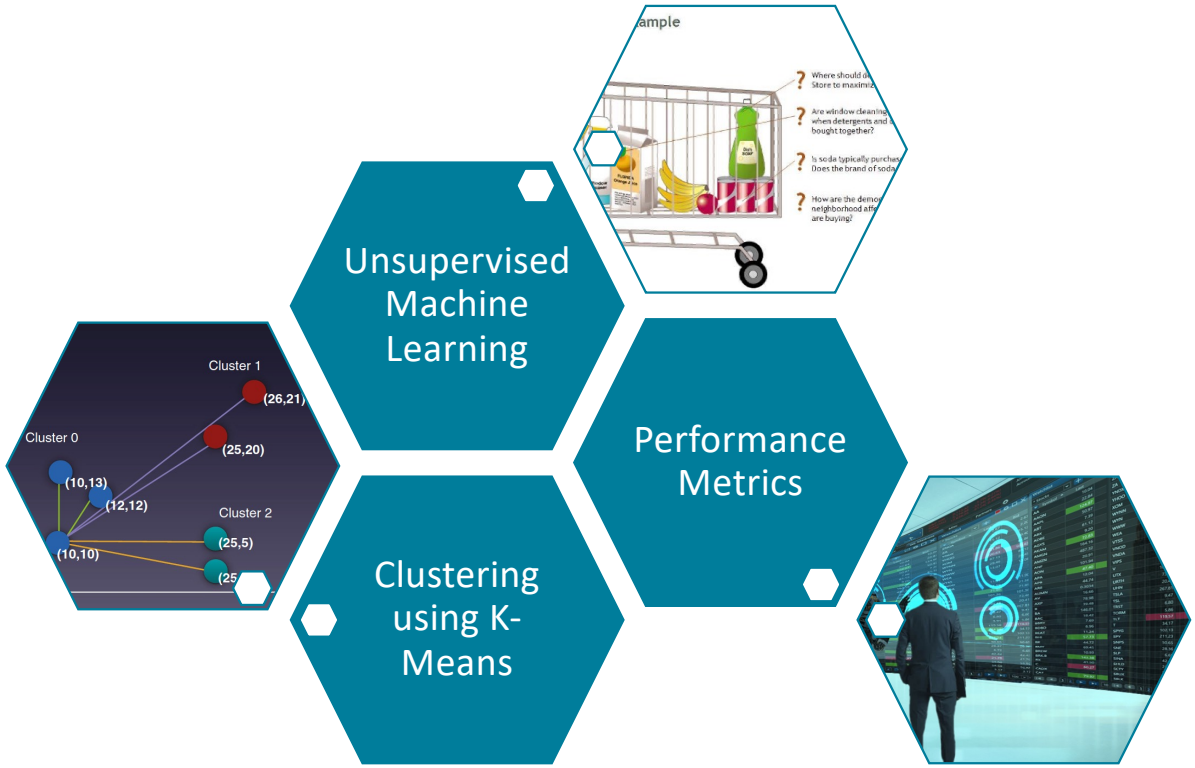
Optimisation
Robust validation

Overview of the Supervised Machine Learning process



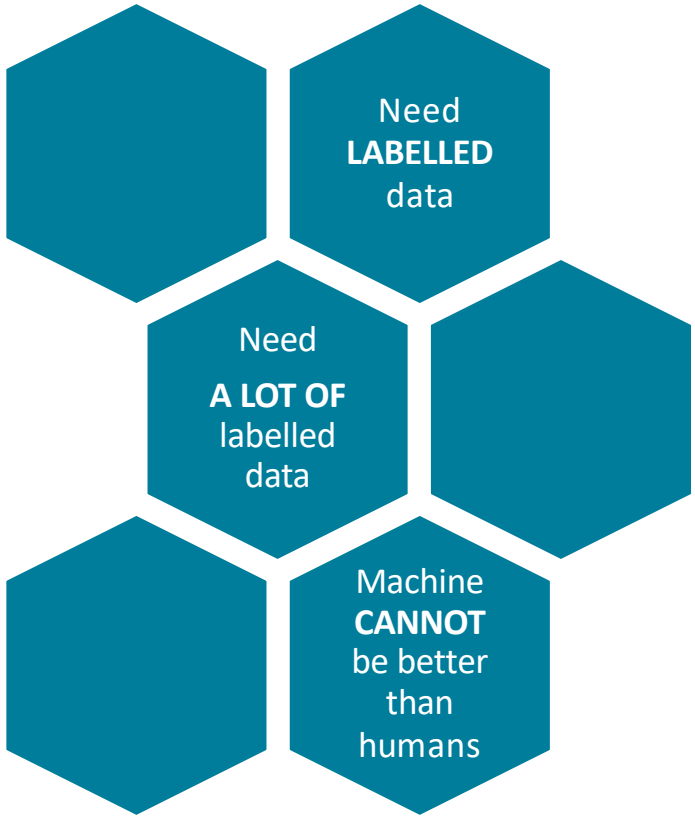
Note: Lab code outlines

Develop, Evaluate and Compare multiple models


Unsupervised Machine Learning
Performance Metrics
Clustering using K-Means
Cluster 1
(26,21)
(25,20)
Cluster 0
(10,13)
(12,12)
Cluster 2
(25,5)
(10,10)

Unsupervised Machine Learning

Problems with supervised machine learning



Machine Learning

Kotu and Deshpande, 2019, chapter 1

Supervised machine learning

- infer a function or relationship based on labelled training data
- use this function to map new unlabelled data to predict output variables
- classification and estimation

Unsupervised machine learning

- there are no output variables to predict
- uncovers hidden patterns in unlabelled data based on the relationships and patterns in the data
- For example, clustering

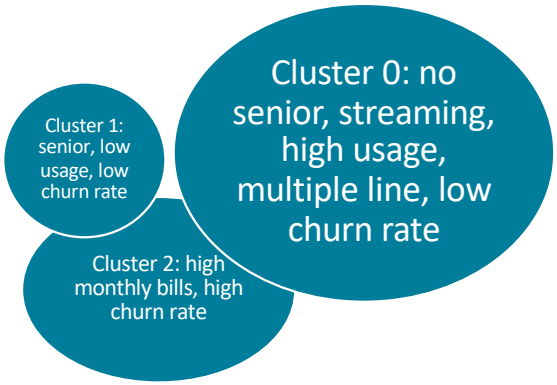
churn probability

loyal probability

ExampleSet (7043 examples, 0 special attributes, 11 regular attributes) Filter (7,043 / 7,043 examples): all

Row No.	customerID	gender	SeniorCitiz...	PhoneServi...	MultipleLin...	TechSupport	Streaming...	PaymentMe...	MonthlyCh...	TotalCharg...	Churn
484	5168-MQQCA	Female	0	Yes	Yes	Yes	Yes	Bank transfe...	108.500	8003.800	No
485	5949-XIKAE	Female	0	Yes	No	No	No	Electronic ch...	83.550	680.050	Yes
486	7971-HLVXI	Male	0	Yes	Yes	No	Yes	Credit card ...	84.500	6130.850	No
487	9094-AZPHK	Female	0	Yes	Yes	No	Yes	Electronic ch...	100.150	1415	No
488	3649-JPUGY	Male	0	Yes	Yes	Yes	Yes	Bank transfe...	88.600	6201.950	No
489	4472-LVYGI	Female	0	No	No phone s...	Yes	No	Bank transfe...	52.550	?	No
490	8372-JUXUI	Male	0	Yes	Yes	No	No	Electronic ch...	74.350	74.350	Yes
491	3552-CTCYF	Male	0	Yes	Yes	No	Yes	Bank transfe...	104.800	6597.250	No
492	6778-YSNIH	Female	0	Yes	No	No	No	Electronic ch...	59	114.150	No
493	0388-EOPEX	Female	0	Yes	No	No	No	Electronic ch...	74.400	139.400	Yes
494	5756-OZRIO	Male	1	Yes	Yes	No	Yes	Bank transfe...	64.050	3902.600	No
495	6579-JPICP	Male	0	Yes	No	No internet ...	No internet ...	Mailed check	20.400	20.400	No
496	8205-OTCHB	Male	0	No	No phone s...	No	Yes	Bank transfe...	43.750	903.600	Yes

https://www.kaggle.com/pavanraj159/telecom-customer-churn-prediction

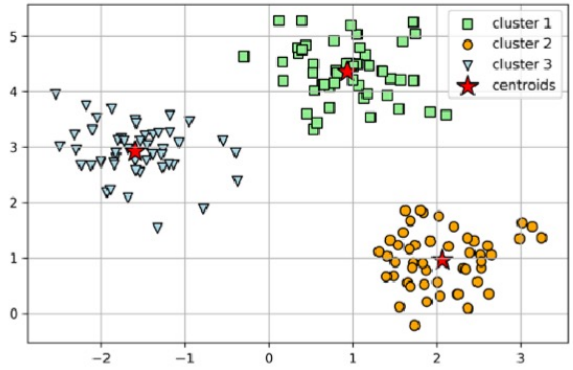


Business applications

- **Customer segmentation**: targeted marketing, personalised recommendations, and tailoring product offerings.
- **Market research**: Analysing customer surveys with many questions (features) to identify the primary factors affecting customer decisions.
- **Market basket analysis:** product placement, promotion, recommendations, guiding cross-selling strategies.
- **Anomaly detection**: By identifying standard groupings, anomalies or outliers can be detected, which can be especially useful in fraud detection or network security.
- **Logistics and Supply Chain**: Optimise routes or warehouse layouts by clustering delivery points or products.
- **Healthcare**:
  - Grouping patients based on symptoms, genetics, or treatment responses for better diagnostic or treatment strategies.
  - Detecting frequently co-occurring diseases or conditions in patient datasets.

Unsupervised Machine Learning problems

- **Clustering**: identify natural groupings in the data without any label or prior knowledge of the groupings.
- **Dimensionality reduction** – principal component analysis: reduce the number of variables without losing important information.
- **Anomaly detection**: detect outliers (data points) that are significantly different from the norm.
- **Association rule learning**: identifying relationships and dependencies between variables - how the variables are related to each other.



Raschka , and Vahid Mirjalili, 2019

ML in Business Framing: Customer Segmentation

- Telco company, high competition in the market
- Customers have difference preferences and needs
- Find customer segments to facilitate create targeted engagement and promotion

Business needs

- Sources:
  - Internal data: contract and transaction data
- **Data content**:
  - Customer demographics and services signed up
  - Behaviour data including monthly usage and total payment

Data

- **Unsupervised ML, clustering**
- Build and test a **K-means clustering** model to inform **targeted engagement and promotion**

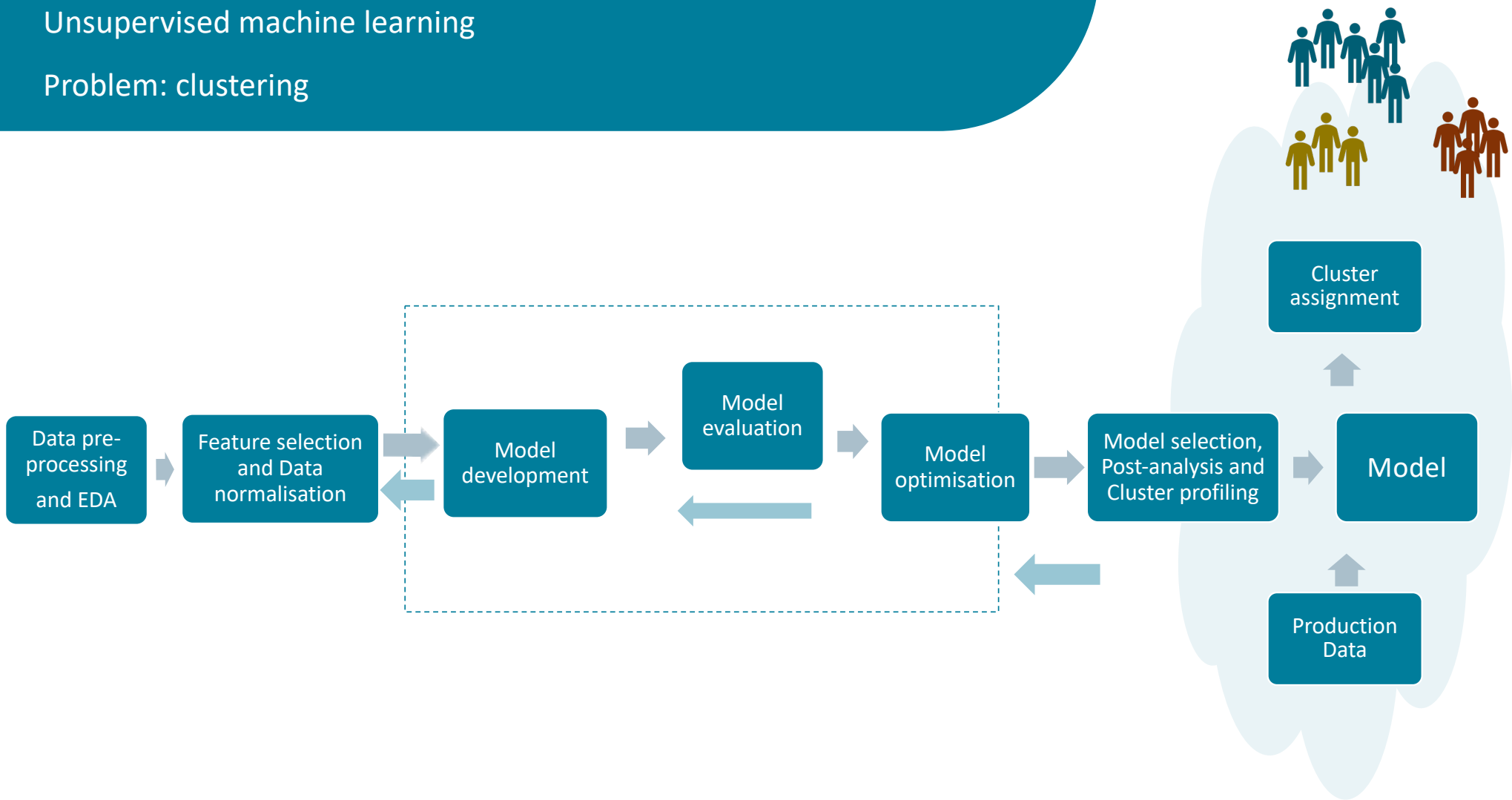
Analytics (ML)

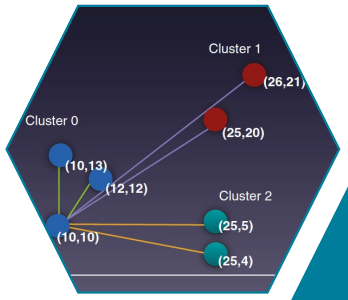
- Targeted engagement and promotion

Deployment

Unsupervised machine learning

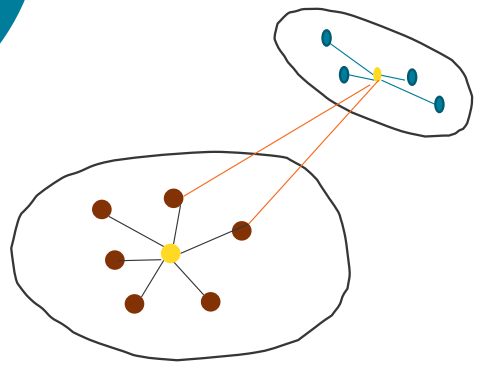
Problem: clustering





Clustering using K-Means

Clustering using K-Means

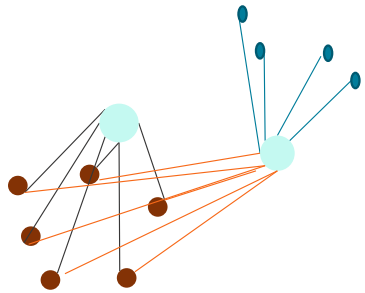


- Objective: Group data points with similar characteristics into k groups (clusters)
- Prototype-based clustering means that each cluster is represented by a prototype, i.e., cluster centres or centroids
- Hard clustering: each data point is exclusively assigned to one cluster.

Mathematical formulation of K-means clustering

- Minimising the sum of squared distances between data points and their assigned cluster centres.
- The algorithm iteratively updates the cluster centres until convergence.

Clustering using K-Means



- Objective: Group data points with similar characteristics into k groups (clusters)
- Prototype-based clustering means that each cluster is represented by a prototype, i.e., cluster centres or centroids
- Hard clustering: each data point is exclusively assigned to one cluster.

1. Specify K
2. (Randomly) initiate K centroids (centres of clusters)
3. Assign data points to their nearest centroid to form K clusters
4. Calculate new centroids for each cluster
5. Repeat steps 3 and 4 until no further change in assignment - i.e. the centroids converge

Example of K-mean simulators

https://vis.yalongyang.com/clustering-vis/index.html

Clustering using K-Means (cont.)

- Use Euclidian distance between datapoints x and y in a m-dimensional space:

$$d(x,y)^2 = \sum_{j=0}^{m-1}(x_j - y_j)^2$$ *j – dimension*

- Optimisation of sums of squared errors, that is, the sum of the squared Euclidean distances of each point to its closest centroid, also called the cluster inertia:

$$SSE = \sum_{i=0}^{n-1}(x_i - c)^2$$ *i – data point in the cluster*

- Objective function J:

$$J = \sum_{j=0}^{k-1}\sum_{i=0}^{n-1}(x_{ji} - c_j)^2$$ *i – data point in the cluster j*
j - cluster

Feature selection for clustering

Do Not Include a Target Variable: If your goal is to create clusters that can be later used for predictive modelling, you should not include the target variable in your clustering process. This is particularly important in a "cluster-then-predict" approach. Including the target variable in clustering can lead to data leakage because you're essentially using information computed from the target variable to predict itself.

Include Target Variable: If your aim is to understand the inherent characteristics of data points without any consideration for predictive modelling, then including the target variable is acceptable. For example, you might want to develop different customer retention or upsell strategies for different customer segments without necessarily predicting anything.

You can choose either of these scenarios based on your specific aims. The key is to have a clear understanding of what you are trying to achieve, interpret the results correctly, and conduct post-analysis and interpretation appropriately.

Case One: Customer Segmentation

```
#loading data
records = pd.read_csv('https://raw.githubusercontent.com/VanLan0/MIS710/main/Customers.csv')
```

- Cleans data as needed
- Convert data into numerical as needed
- Let's do a sample clustering based on two features
- Scale data to ensure that the selected features have the same scale

```
# Select relevant features for clustering
features=['MonthlyCharges', 'tenure']
X = records[features]
```

- ❖ The choice of features should be explained.
- ❖ The choice to include or exclude the Churn variable should align with the specific analytical objectives.
- ❖ Scaling is needed.

Case One: Customer Segmentation

- **Model development**
- Model evaluation
- Model optimisation

	gender	tenure	MonthlyCharges	Cluster
4413	Female	17	89.15	2
1244	Female	15	19.40	0
6506	Female	13	84.60	2
3832	Female	51	78.65	1
2374	Male	2	19.40	0
756	Male	66	19.70	0

```
from sklearn.cluster import KMeans

# Fit K-means clustering model with optimal number of clusters
k=3
kmeans = KMeans(n_clusters=k, n_init='auto',max_iter=300, random_state=2023)
kmeans.fit(X_scaled)

# Add cluster labels to original dataset
records['Cluster'] = kmeans.labels_


#Print sample rows for selected columns to see the clusters
records[['gender','tenure','MonthlyCharges','Cluster']].sample(10)
```

Case One: Customer Segmentation

- Model development
- **Model evaluation: Within-Cluster Sum of Squares / Sum of Squared Errors**
- Model optimisation

- Within-Cluster Sum of Squares (WCSS) is the sum of the squared distance between each data point and its assigned cluster centroid, averaged over all the clusters.
- WCSS measures the **compactness** of the clusters, with lower values indicating tighter and more compact clusters.
- Sum of Squared Errors (SSE)

$$\sum_{i=1}^{k} \sum_{x \in Ci} ||x - C_i||^2$$

```
# Evaluate the model using within-cluster sum of squares (WCSS)
wcss = kmeans.inertia_
print("Within-Cluster Sum of Squares (WCSS):", '%.3f' % wcss)
```

```
Within-Cluster Sum of Squares (WCSS): 4622.613
```

Case One: Customer Segmentation

- Model development
- **Model evaluation: Davies-Bouldin Index**
- Model optimisation

$$\frac{1}{k}\sum_{i=1}^{k} max_{i \neq j}\left(\frac{S_i+S_j}{D_{ij}}\right)$$

Si = average distance of each point in cluster i to its centroid

- the average of the maximum ratio of the within-cluster distance and the between-cluster distance for each cluster.
- Davies-Bouldin Index measures the similarity between each cluster Ci and its **most similar** cluster Cj
  - The lower the better

0 to 1: Indicates distinct clusters.
Above 1: indicates poor separation between clusters.

```
from sklearn.metrics import davies_bouldin_score

# Compute the Davies-Bouldin index
dbs = davies_bouldin_score(X_scaled, kmeans.labels_)
print("Davies Bouldin index:", '%.3f' % dbs)
```

```
Davies Bouldin index: 0.799
```

Case One: Customer Segmentation

- Model development
- **Model evaluation: Silhouette coefficient** $= 1 - \left(\frac{a}{b}\right)$
- Model optimisation

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

- ***Cohesion****: a, mean intra-cluster distance, is the average distance of a point to all other points in the* ***same cluster***
- ***Separation****: b, nearest-cluster distance, is the lowest average distance of a point to all other points in the* ***closest cluster****; if b is large, cluster separation is good*

- Silhouette score measures the within cluster cohesion (how similar the data points are within each cluster), and between cluster separation (how well-separated the clusters are).
- Ranges from -1 to 1.
  - Close to 1 means that the data points within a cluster are very similar to each other, and very different from the data points in other clusters.
  - 0 means that the data points are equally similar to neighboring clusters.
  - Negative means that the data points may have been assigned to the wrong cluster.

Case One: Customer Segmentation

- Model development
- **Model evaluation**
- Model optimisation

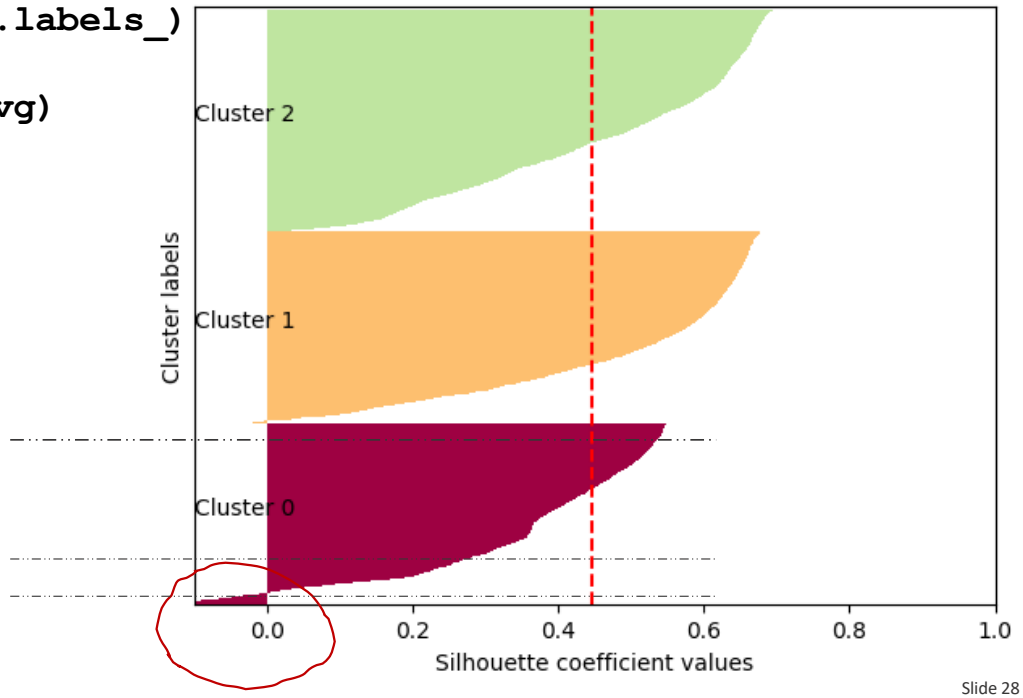
Metric	K	Sample Size	Feature Size	Outliers
WCSS (Within-Cluster Sum of Squares)	Highly sensitive	Somewhat sensitive, more reliable with large sample size	Sensitive, artificially inflating WCSS	Very sensitive, artificially inflating WCSS.
Davies-Bouldin Index	Sensitive, but non-monotonically, too many clusters leads to poor separation	Less sensitive	Sensitive, affected by irrelevant or noisy	Moderately sensitive
Silhouette Coefficient	Sensitive, but non-monotonically, too many clusters leads to poor separation	Not significantly sensitive, by measuring relative distances	Sensitive, affected by irrelevant or noisy	Moderately sensitive

Case One: Customer Segmentation

- Model development
- **Model evaluation: Silhouette coefficient =1-(a/b)**
- Model optimisation

```
# Compute the Silhouette score for the clustering model
silhouette_avg = silhouette_score(X_norm, kmeans.labels_)

print('Silhouette score:', '%.3f' % silhouette_avg)
```

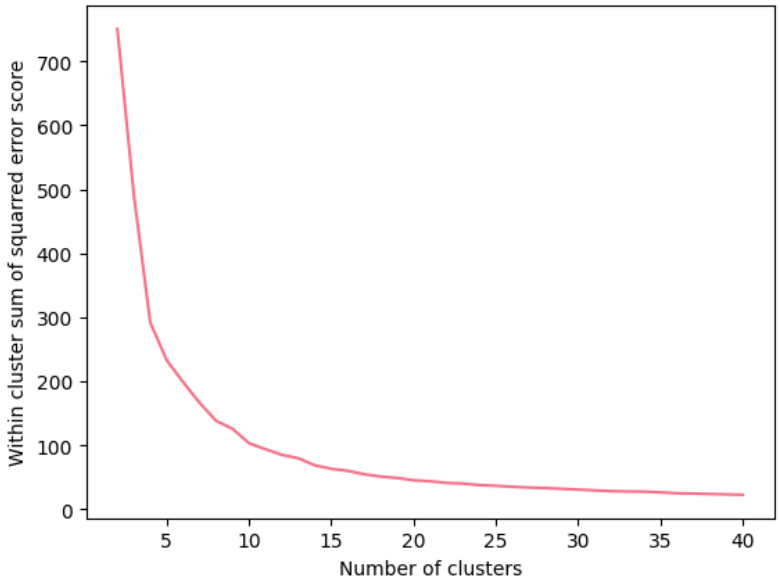
```
Silhouette score: 0.446
```



Case One: Customer Segmentation

- Model building and visualisation
- Model evaluation: Silhouette coefficient
- **Model optimisation: find the best k using the elbow method**

```
# Determine optimal number of clusters using the SSE metric
sse_scores = []
best_k=3
best_sse_score=4622.613
for k in range(2, 21):
    kmeans = KMeans(n_clusters=k, n_init=2)
    kmeans.fit(X_norm)
    sse_score_k=kmeans.inertia_
    sse_scores.append(sse_score_k)
    if best_sse_score > sse_score_k:
        best_k = k
        best_sse_score = sse_score_k

print('Best k: ', best_k)
print('Best within cluster sum of squared error score: ',
'%.3f' %best_sse_score)
```




```
Best k: 20
Best within cluster sum of squared
error score: 457.120
```

Case One: Customer Segmentation

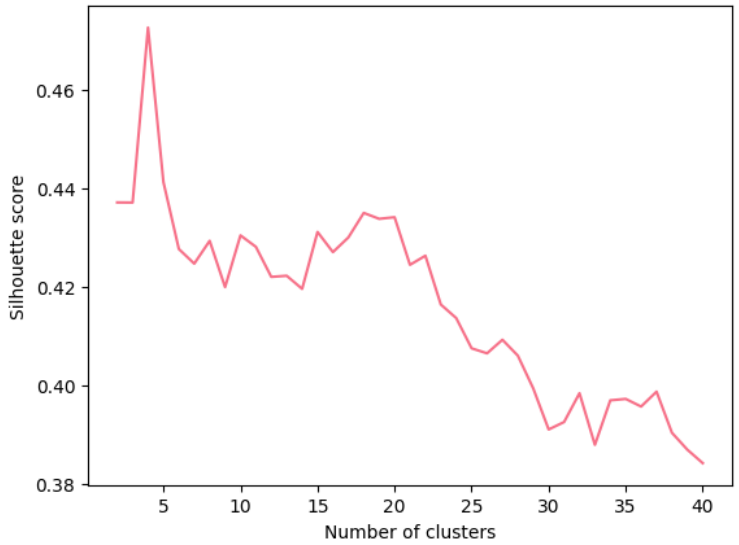
- Model building and visualisation
- Model evaluation: Silhouette coefficient
- **Model optimisation: find the best k using silhouette scores**

```
# Determine optimal number of clusters using silhouette score
sil_scores = []
best_k=2
best_sil_score=0
for k in range(2, 21):
    kmeans = KMeans(n_clusters=k, n_init=2)
    kmeans.fit(X_norm)
    sil_score_k=silhouette_score(X_norm, kmeans.labels_)
    sil_scores.append(sil_score_k)
    if best_sil_score < sil_score_k:
        best_k = k
        best_sil_score = sil_score_k

print('Best k: ', best_k)
print('Best silhouette score: ', '%.3f' %best_sil_score)
```

Case One: Customer Segmentation

- Model building and visualisation
- Model evaluation: Silhouette coefficient
- **Model optimisation: find the best k using silhouette scores**

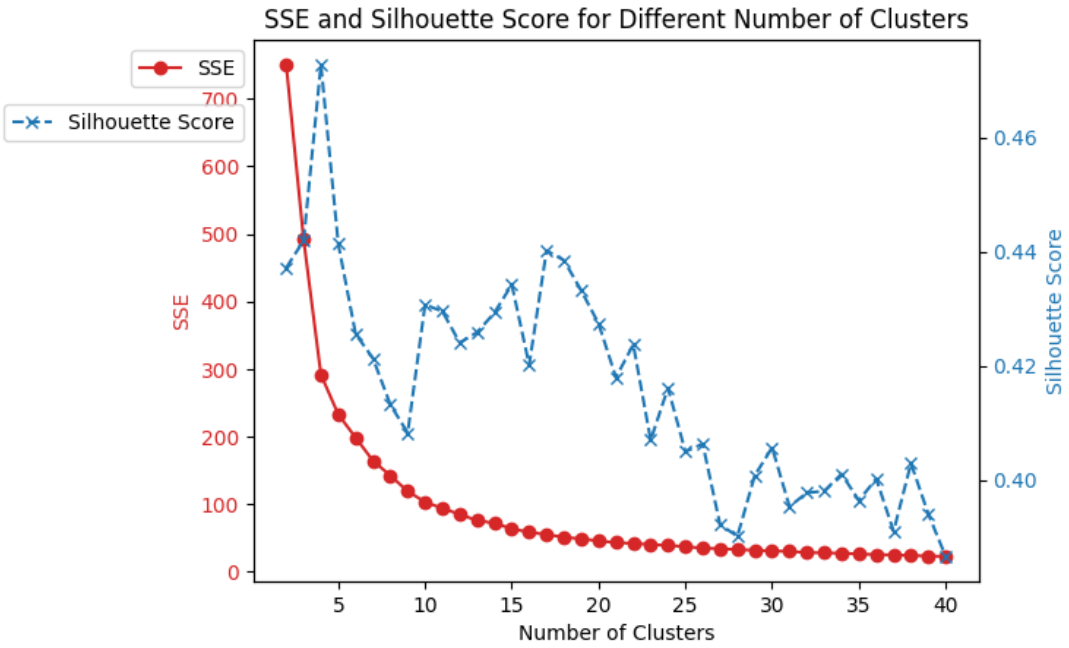
```
# Plot the silhouette scores to determine
optimal number of clusters
plt.plot(range(2,21), sil_scores)
plt.xlabel("Number of clusters")
plt.ylabel("Silhouette score")
plt.show()
```

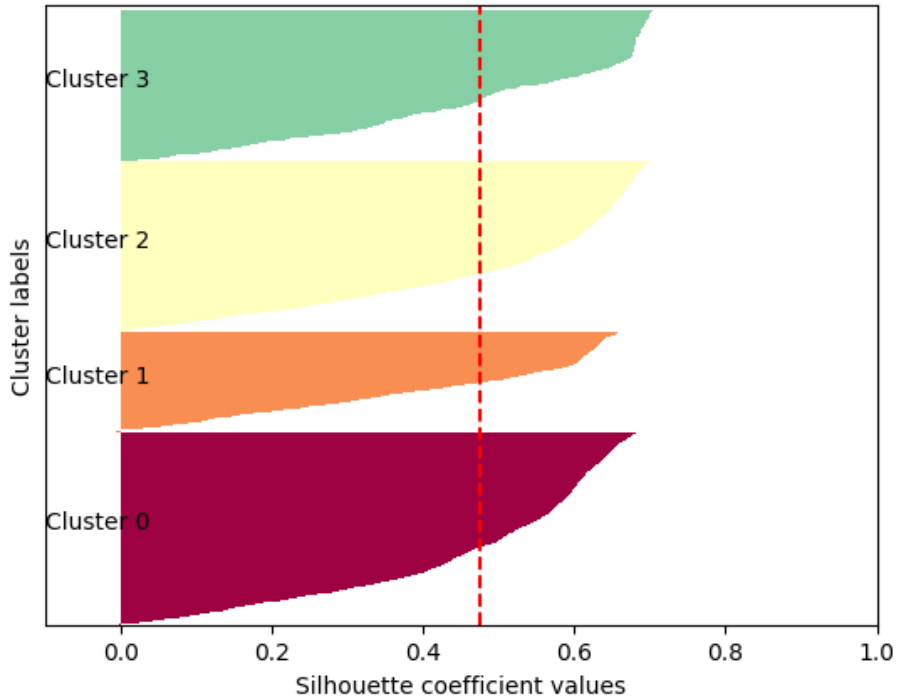



```
Best k: 4
Best silhouette score: 0.476
```

Case One: Customer Segmentation

- Model building and visualisation
- Model evaluation: Silhouette coefficient
- **Model optimisation: find the best k**

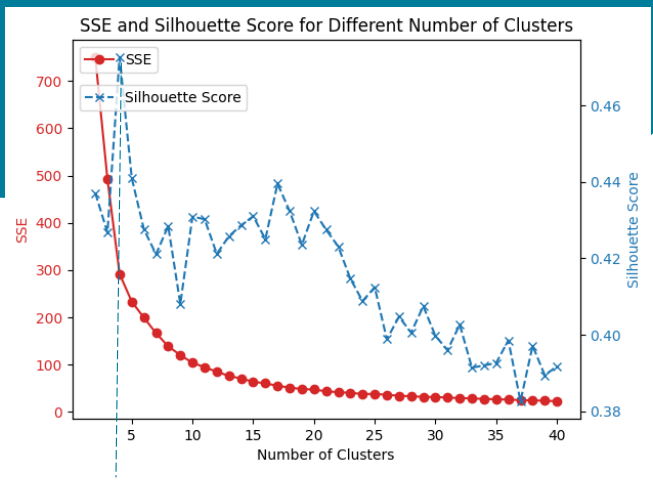


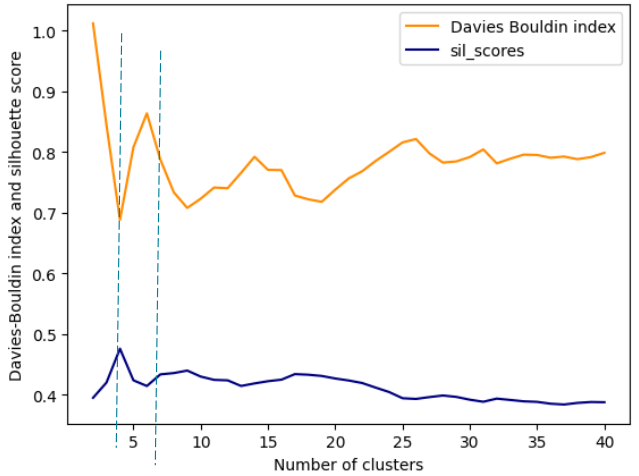


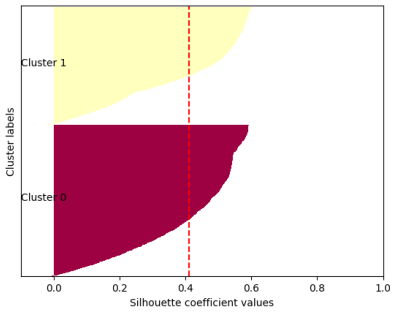

```
Best k: 4
Best silhouette score: 0.476
```

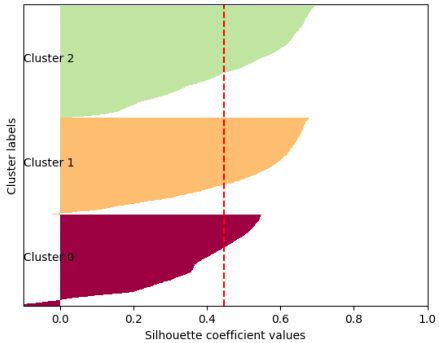
Case One: Customer Segmentation

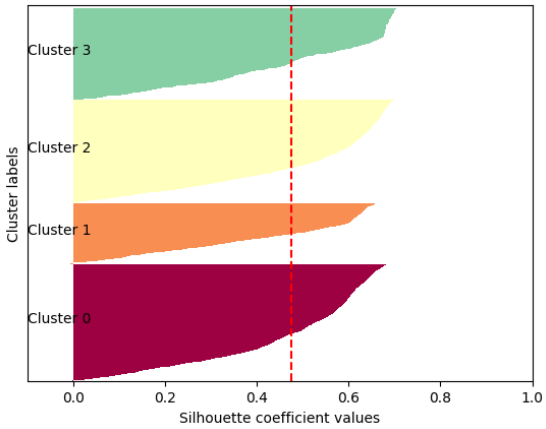
- Model building and visualisation
- Model evaluation:
- **Model optimisation: multiple factors**

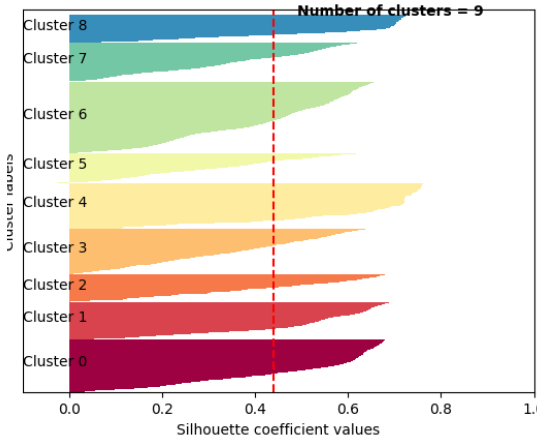


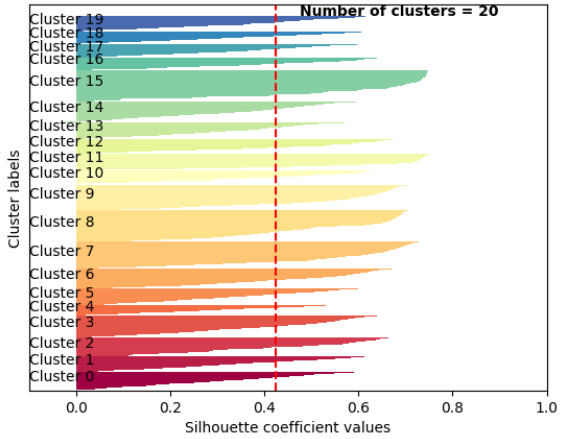








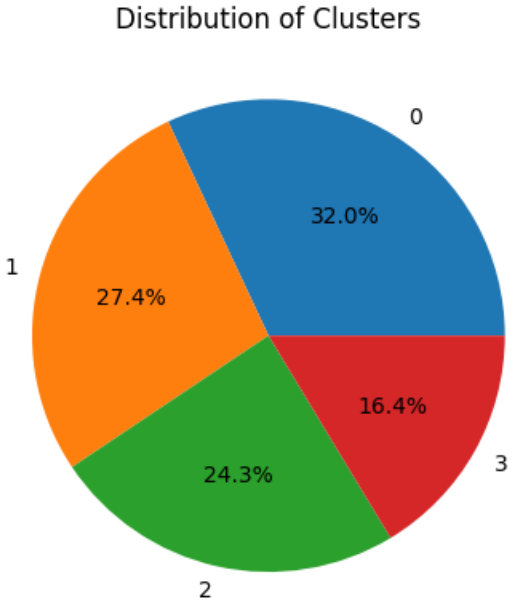


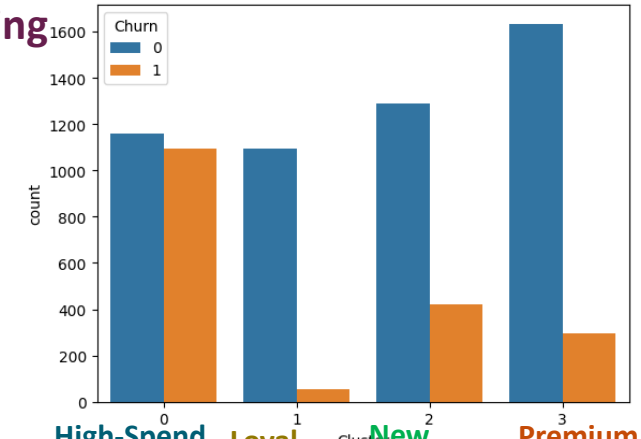


- Domain expertise, Business value and Computation cost

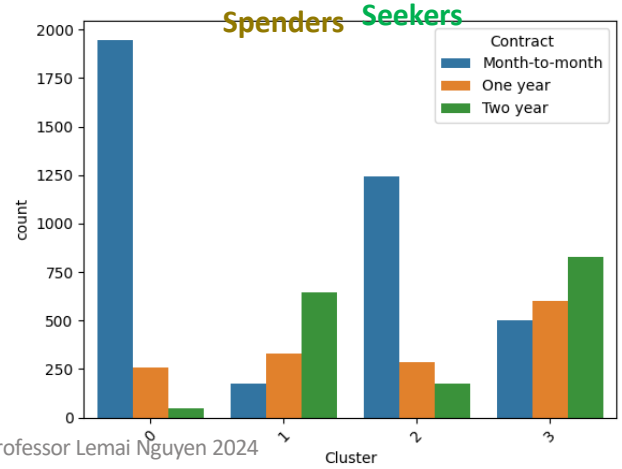
Case One: Customer Segmentation

- **Post-analysis & Cluster Profiling**

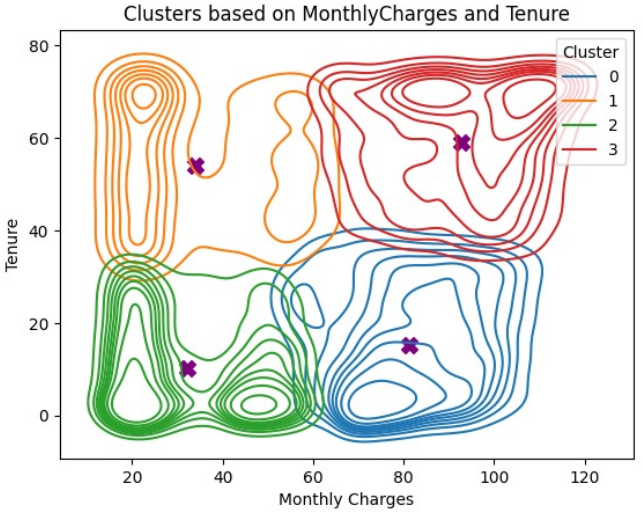


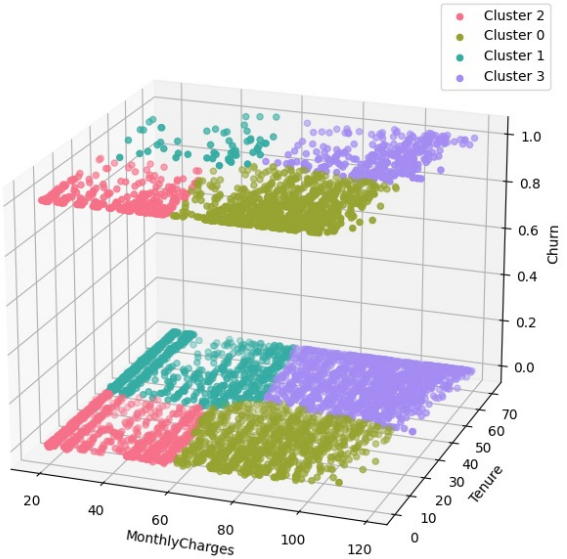


High-Spend Drifters" **Loyal Low-Spenders** **New Budget Seekers** **Premium Loyalists**



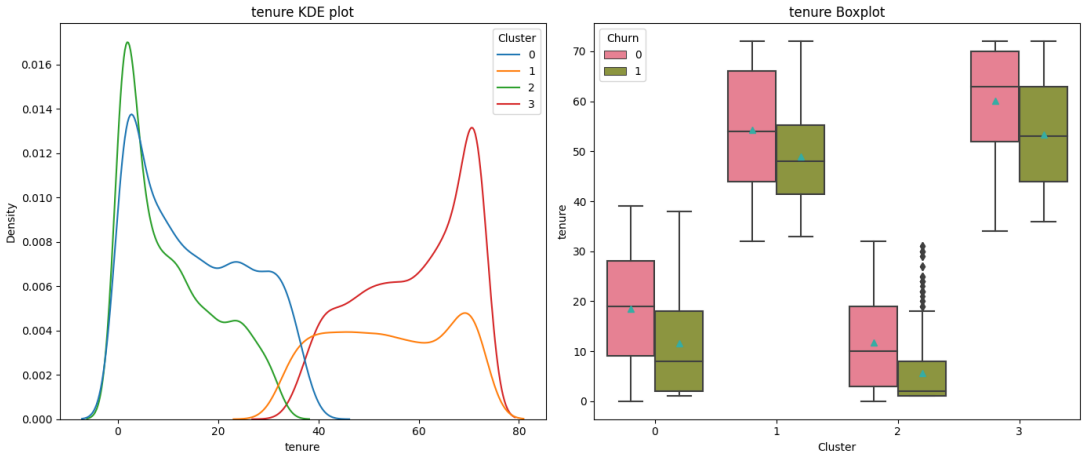
Name/label clusters

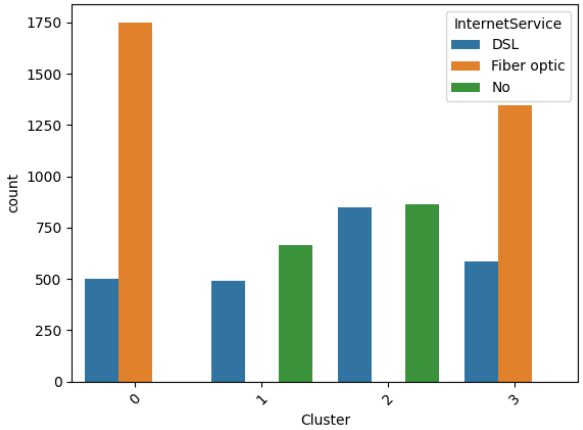


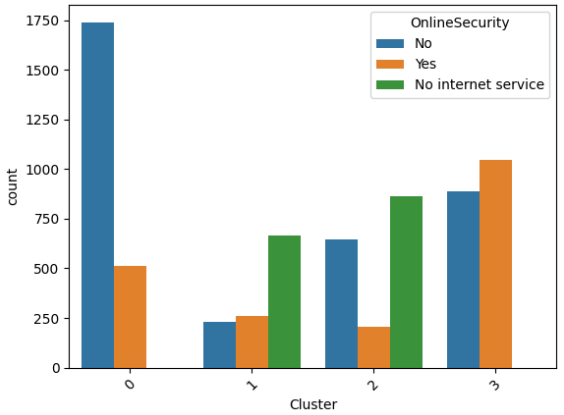


Label/name clusters

- **Post-analysis & Cluster Profiling**



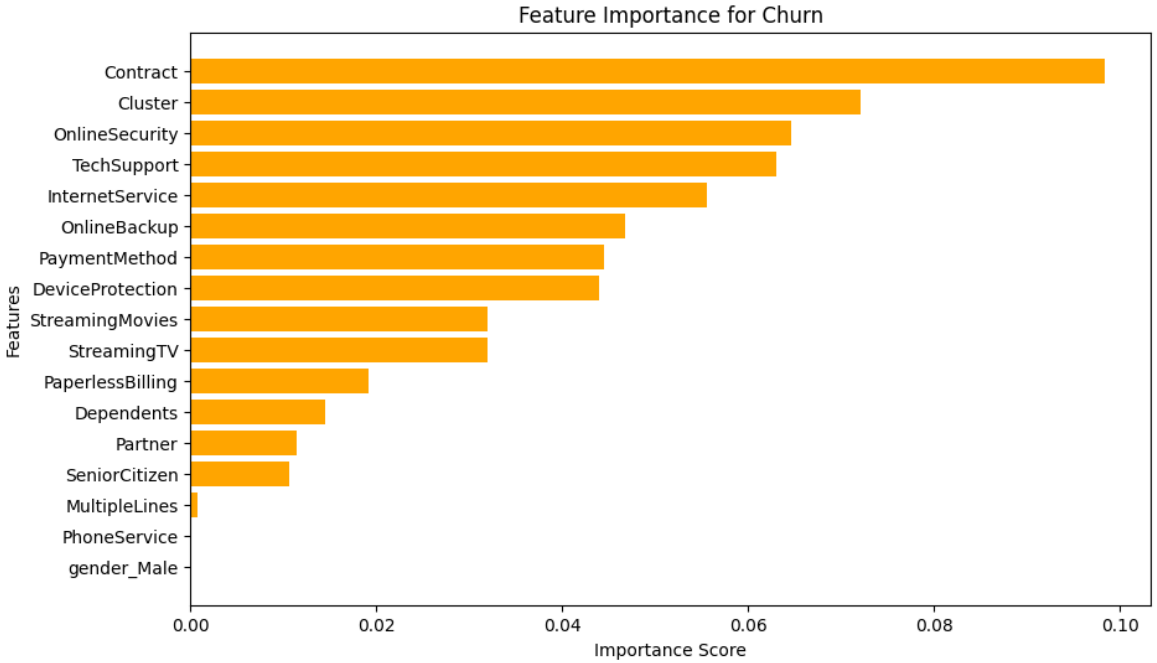


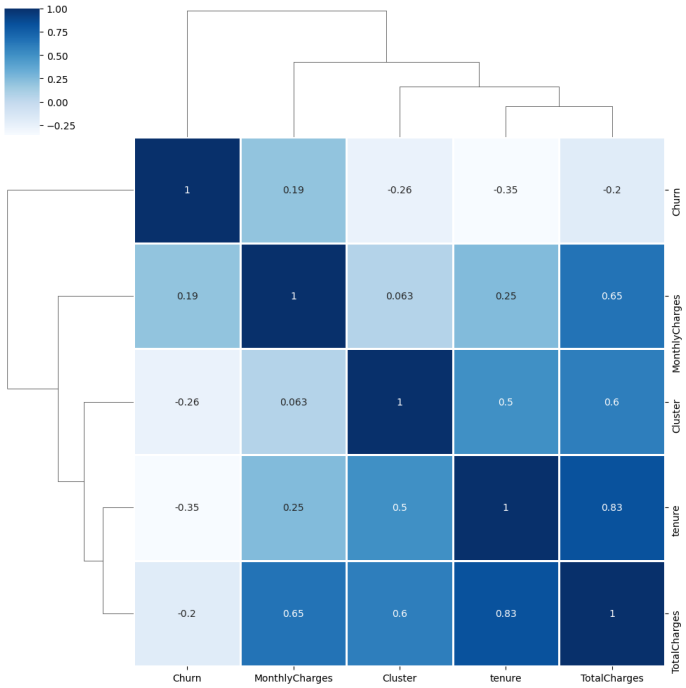


Don’t include Churn as a feature, then explore which variables can serve as good predictors for each Cluster?

Case One: Customer Segmentation

- **Post-analysis & Cluster Profiling**

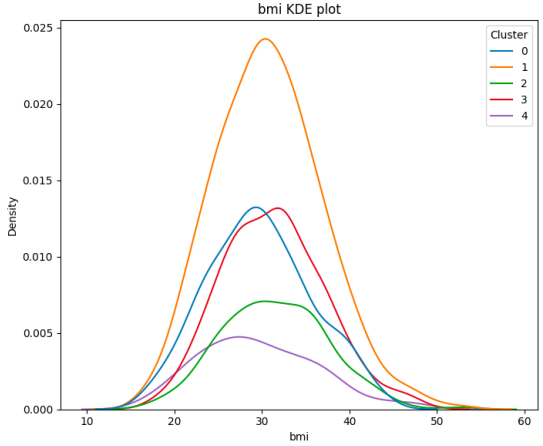


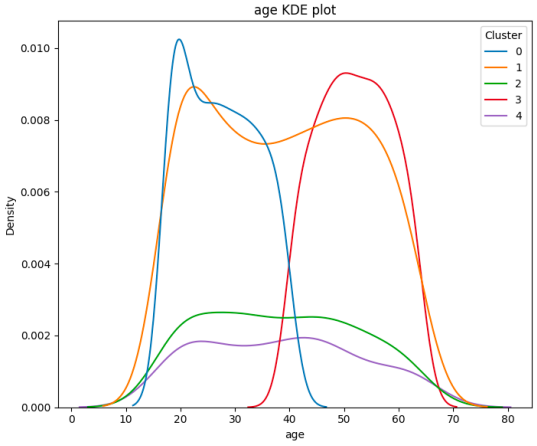


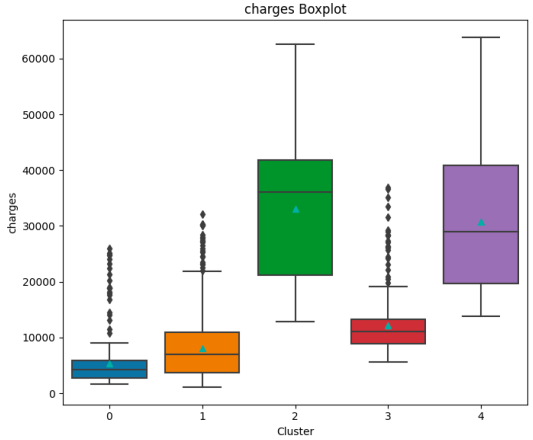
Don’t include Churn as a feature, then explore which variables can serve as good predictors for each Cluster?

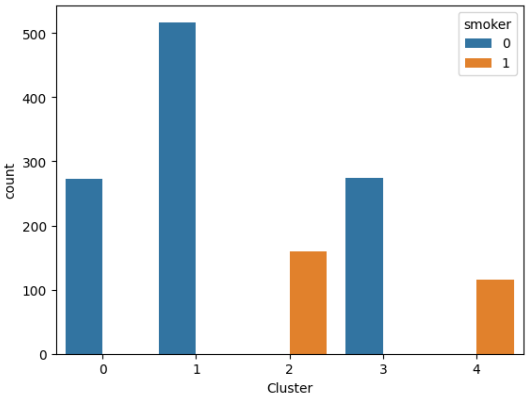
Case Two: Health Insurance Cost Clustering

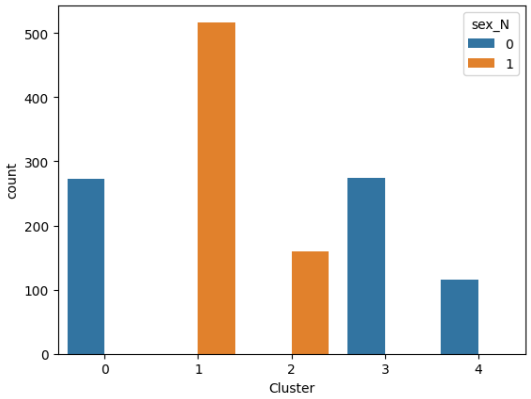
*Include **charges**, smoker, age, sex_N (1 - male), and bmi*

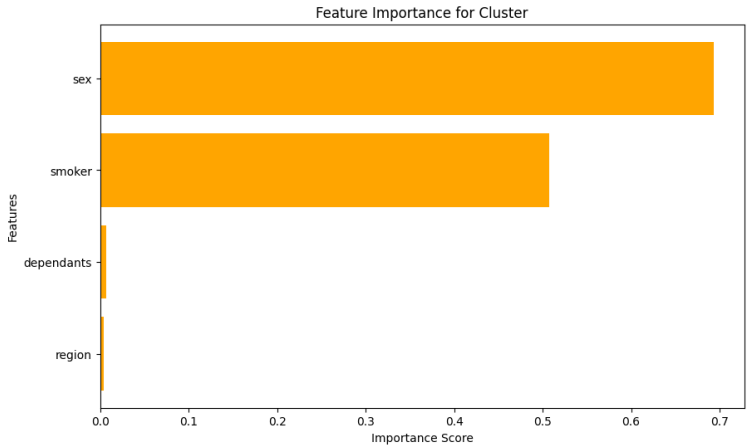








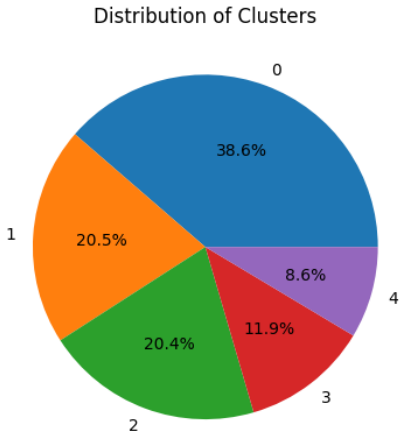


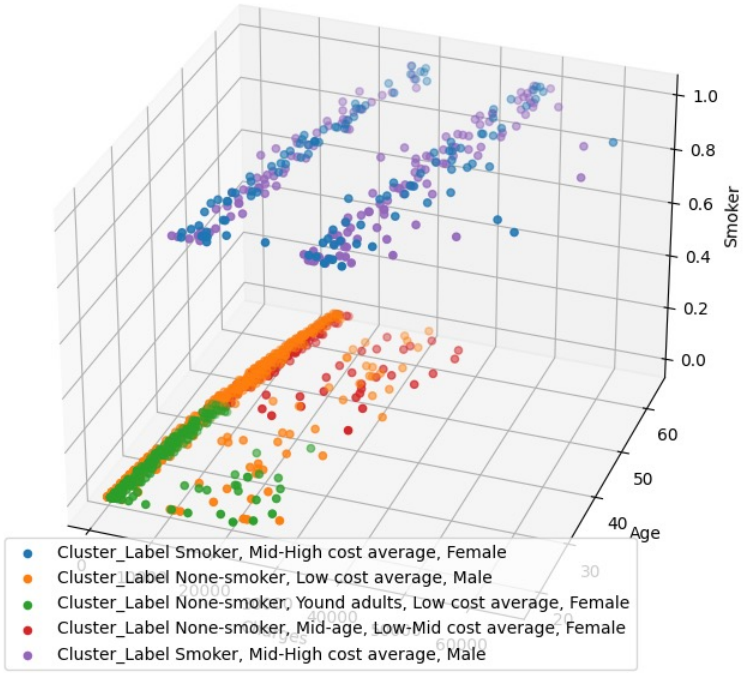


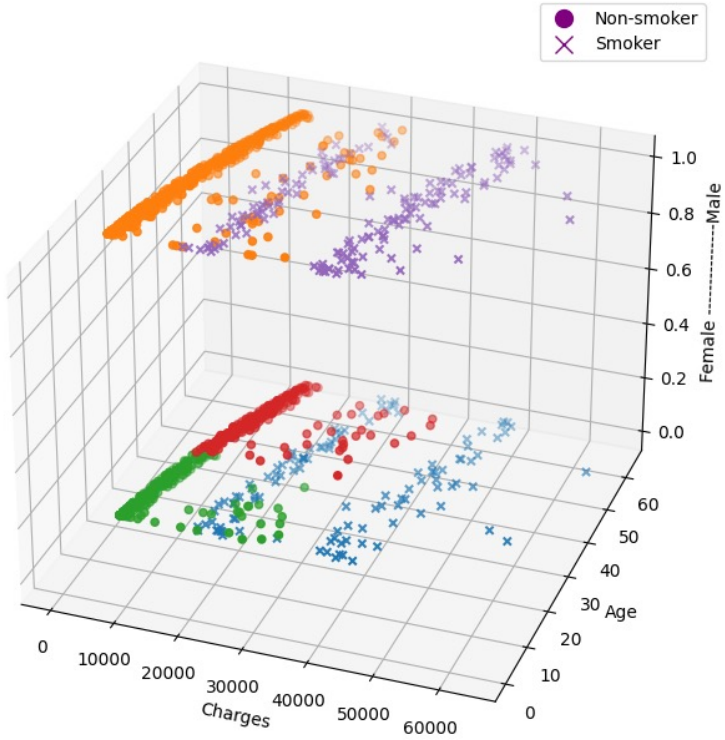
Case Two: Health Insurance Cost Clustering

*Include **charges**, smoker, age, sex_N (1 - male), and bmi*

Label/name clusters







Identify characteristics of cost clusters to design interventions to better manage risk and inform insurance package personalisation.

Note: Avoid Data Leakage – don't use clusters to predict charges!

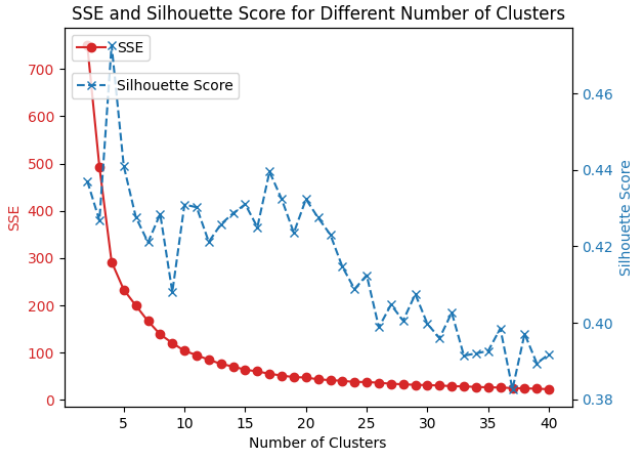
K-Means clustering discussion points

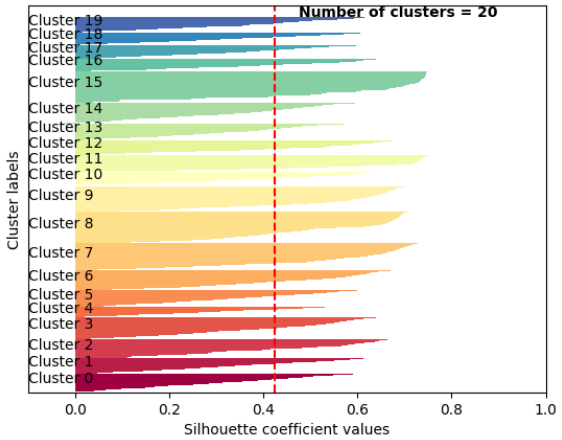
Pros

- Simple and intuitive
- Efficiency (n*k*iterations)
- Works well for spherical clusters

Cons

- **Need to specify k**
- Different initialisations result in different clusters
- Sensitive to outliers
- Sensitive to feature processing
- The curse of dimensionality
- Not effective with complicated geometric shapes
- Validity of clusters
  - Expert judgment is needed
  - AI Ethics is important! See the extended FAT framework – Topic 8





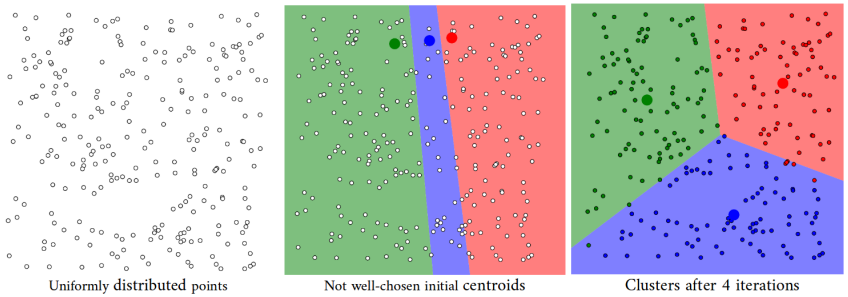
K-Means clustering discussion points

Pros

- Simple and intuitive
- Efficiency (n*k*iterations)
- Works well for spherical clusters

Cons

- Need to specify k
- **Different initialisations result in different clusters**
- Sensitive to outliers
- Sensitive to feature processing
- The curse of dimensionality
- Not effective with complicated geometric shapes
- Validity of clusters
  - Expert judgment is needed
  - AI Ethics is important!



https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

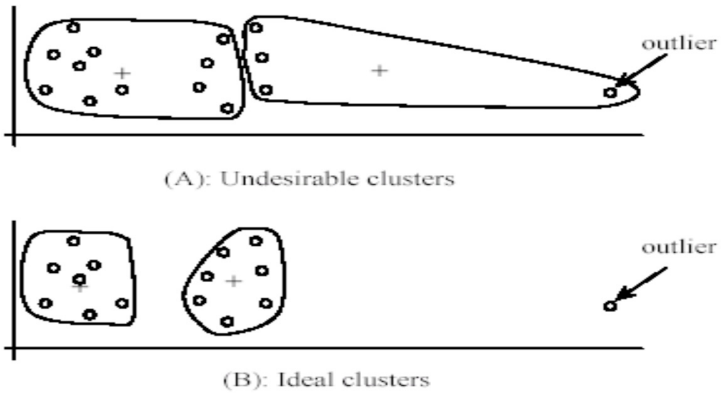
K-Means clustering discussion points

Pros

- Simple and intuitive
- Efficiency (n*k*iterations)
- Works well for spherical clusters

Cons

- Need to specify k
- Different initialisations result in different clusters
- **Sensitive to outliers**
- Sensitive to feature processing
- The curse of dimensionality
- Not effective with complicated geometric shapes
- Validity of clusters
  - Expert judgment is needed
  - AI Ethics is important!



http://www.mit.edu/~9.54/fall14/slides/Class13.pdf

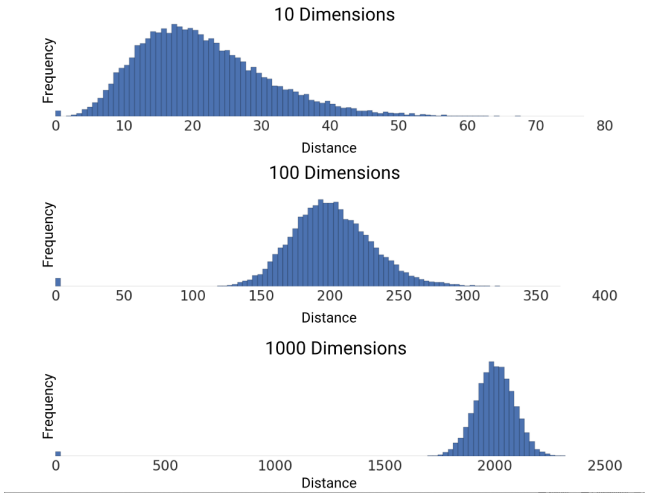
K-Means clustering discussion points

Pros

- Simple and intuitive
- Efficiency (n*k*iterations)
- Works well for spherical clusters

Cons

- Need to specify k
- Different initialisations result in different clusters
- Sensitive to outliers
- **Sensitive to feature processing**
- **The curse of dimensionality**
- Not effective with complicated geometric shapes
- Validity of clusters
  - Expert judgment is needed
  - AI Ethics is important!



https://developers.google.com/machine-learning/clustering/algorithm/advantages-disadvantages

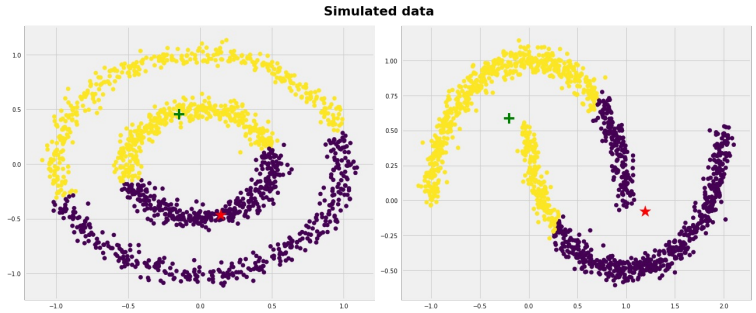
K-Means clustering discussion points

Pros

- Simple and intuitive
- Efficiency (n*k*iterations)
- Works well for spherical clusters

Cons

- Need to specify k
- Different initialisations result in different clusters
- Sensitive to outliers
- Sensitive to feature processing
- The curse of dimensionality
- **Not effective with complicated geometric shapes**
- Validity of clusters
  - Expert judgment is needed
  - AI Ethics is important!



https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a

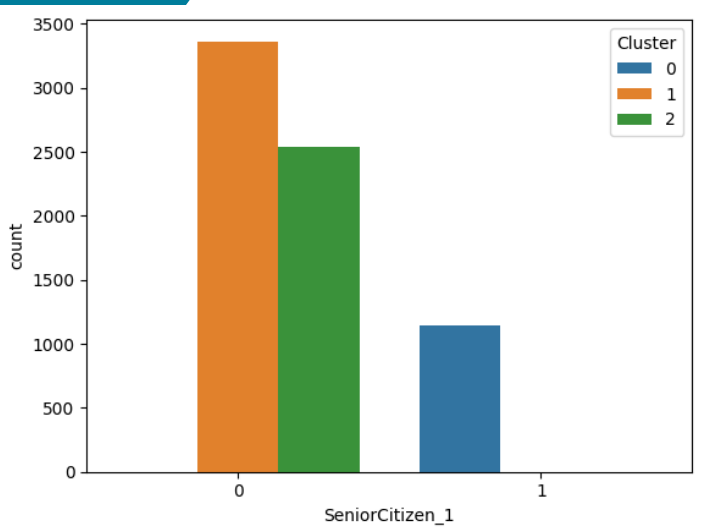
K-Means clustering discussion points

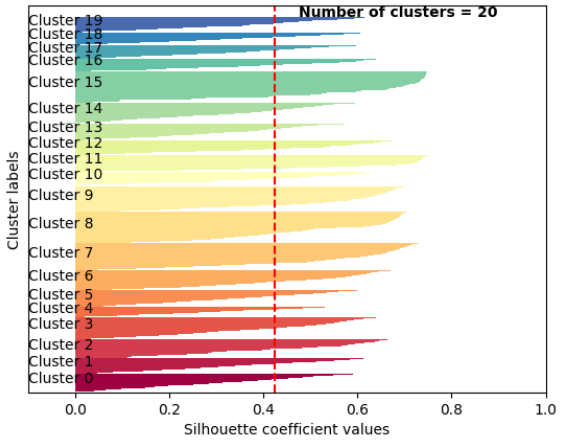
Pros

- Simple and intuitive
- Efficiency (n*k*iterations)
- Works well for spherical clusters

Cons

- Need to specify k
- Different initialisations result in different clusters
- Sensitive to outliers
- Sensitive to feature processing
- The curse of dimensionality
- Not effective with complicated geometric shapes
- **Validity of clusters**
  - Expert judgment is needed
  - AI Ethics is important!





K-Means clustering discussion points

- Normalise data since K-means is distance based
- Consider multiple metrics to optimise K
- Improve initialisation algorithms
- Run the process multiple times and evaluate
- Domain expertise is needed to verify clusters
- Good clusters can be used to supervise classification

K-Medoids – A medoid is an actual data point in the cluster that is the most centrally located.

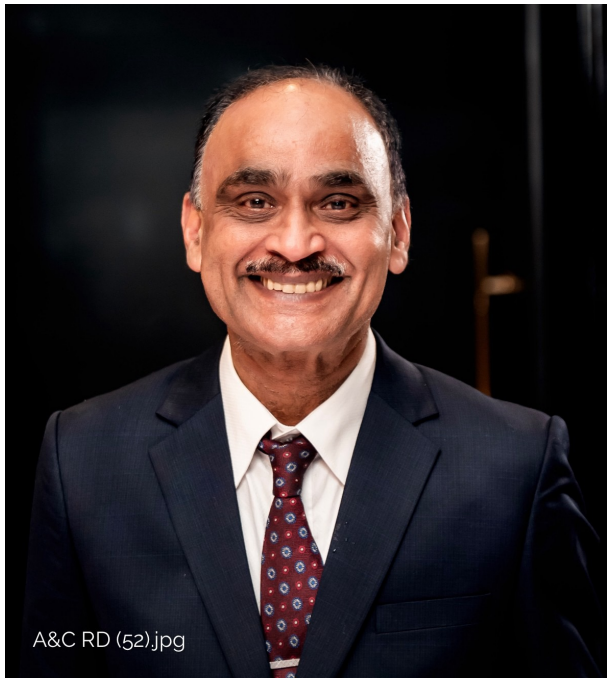
- minimising sum of the pairwise distances between objects
- more robust and flexible clustering algorithm
- handle different types of distances
- less sensitive to outliers
- it can be slower and less computationally efficient than K-Means.

K-means clustering - Recap

- Clustering aims to find groups of ‘similar’ data points based on selected features
  - K-means is prototype based
    - Distance based
    - Sensitive to K, outliers, initialisations, complex data shapes
    - Optimisation of k
  - Density based clustering: Intro to DBSCAN intuition
    - Intro to HDBSCAN intuition
  - Post-hoc analysis and Cluster profiling and labelling
- There are other types of unsupervised machine learning: PCA, anomaly detection, associate rules learning
- Domain expertise is needed
- Like supervised ML, AI ethics is important

Dr. Raju Varanasi

Professor of Practice



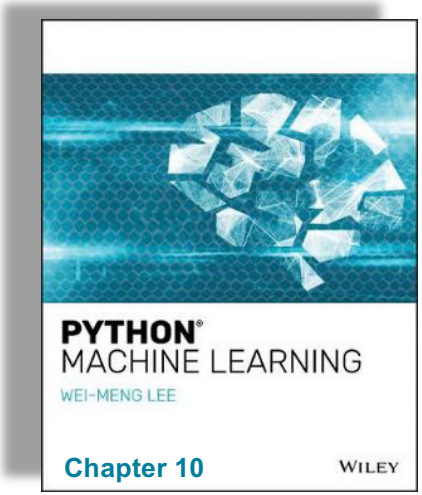
Dr. Raju Varanasi commenced as Professor of Practice in the Department of Information systems and Business Analytics at the Deakin Business School. Dr. Varanasi brings a distinguished blend of educational excellence and industry expertise with a career that spans over three decades in Australia.

Dr. Varanasi has held pivotal roles in TAFE NSW, public and Catholic schools in leveraging digital technologies and data analytics for transformational impact. Dr. Varanasi is a Chemical Engineer an MBA from IIT, New Delhi and IIM Bangalore- world class universities in India. An Australian Fulbright Scholar, Dr. Varanasi attained his PhD from University of Newcastle with his thesis on transforming school systems. His visionary approaches to education have been recognized with awards such as the Top 50 Australian CIO Award (twice) and the Gartner Innovation in Education Award.

He is also a published author, contributing to the literature on educational analytics and school progress frameworks. After roles as a General Manager, Director, COO and CIO in the last 20 years, Dr. Varanasi is consulting for Google's partners across Asia Pacific advising on data, analytics & AI strategies. As a Professor of Practice, Dr. Varanasi is committed to forging stronger partnerships between academia and industry, preparing students to navigate and excel in the dynamic field of Analytics & AI to equip them with the skills necessary to lead enterprises in a data-driven future.

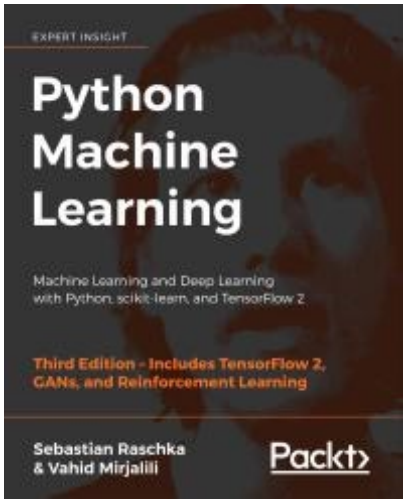
Dr. Raju Varanasi | LinkedIn

From your books



Chapter 10

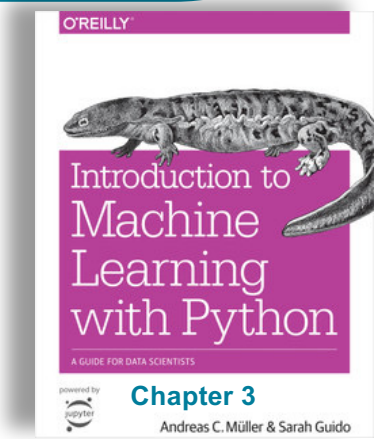
https://ebookcentral-proquest-com.ezproxy.b.deakin.edu.au/lib/deakin/detail.action?docID=5747364



Chapter 11

Raschka and Mirjalili, 2019

https://ebookcentral-proquest-com.ezproxy-b.deakin.edu.au/lib/deakin/reader.action?docID=6005547&ppg=43



Introduction to Machine Learning with Python
Andreas C. Müller and Sarah Guido (2016)

https://learning.oreilly.com/library/view/introduction-to-machine/9781449369880/?sso_link=yes&sso_link_from=Deakin

See useful sites in Lab 9

AND

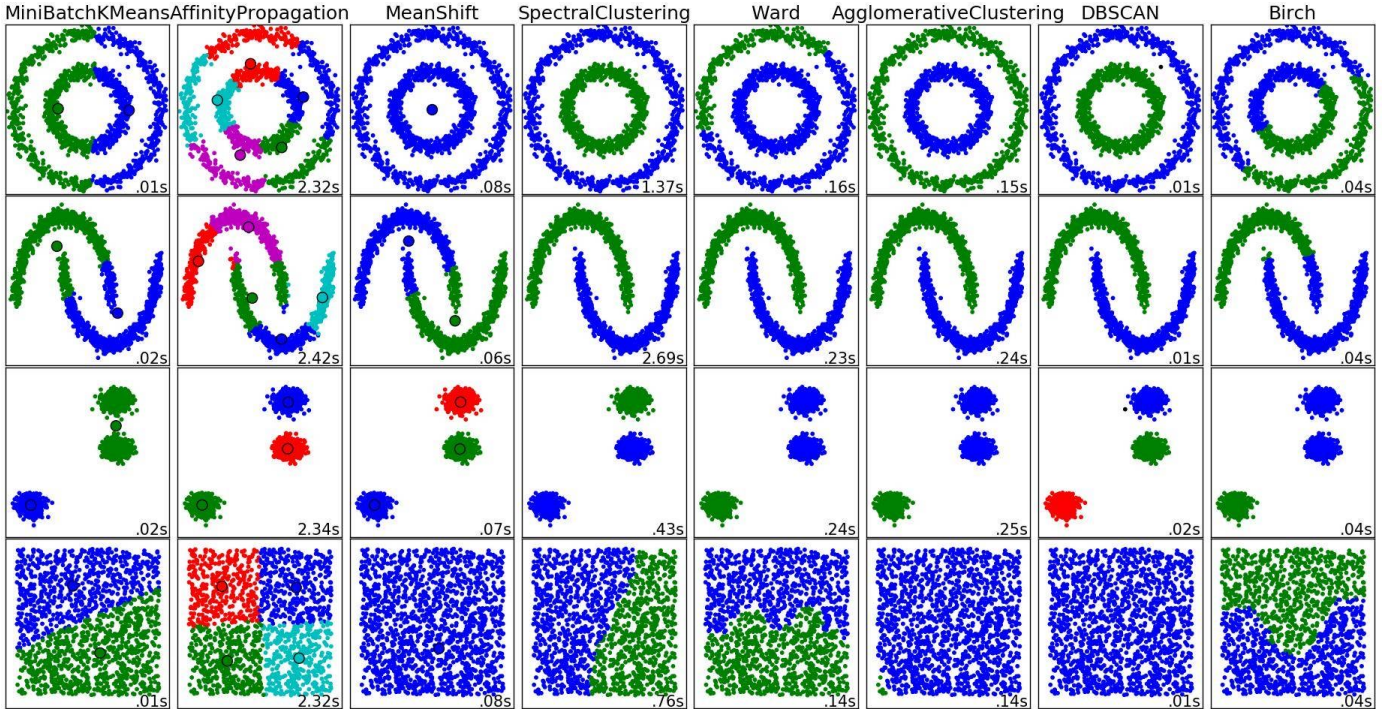
Self-study

Soft clustering

- Also known as fuzzy clustering, assigns each data point to multiple clusters with membership probability scores
- Examples:
  - Fuzzy C-means clustering (FCM): an extension of K-means that allows for overlapping clusters.
    - Possibilistic C-means clustering (PCM): more flexible than FCM in the assignment of membership scores.
  - Gaussian Mixture Models (GMM): assuming the data is generated from a mixture of Gaussian distributions – can apply to hard or soft clustering

Alternative algorithms

- Clustering: Groups similar data points together
- Density-based: Focuses on areas with high data point density
- Hierarchical: Represent data relationships using a tree-like structure



Density-based spatial clustering of applications with noise (DBSCAN)

- **Directly Reachable**: Point X is directly-reachable from point A if it is within a radius ε to A:

$$d(A, X) \leq \varepsilon$$

- **Core points:** Point A becomes a core point if it is surrounded by at least n_{min} points:

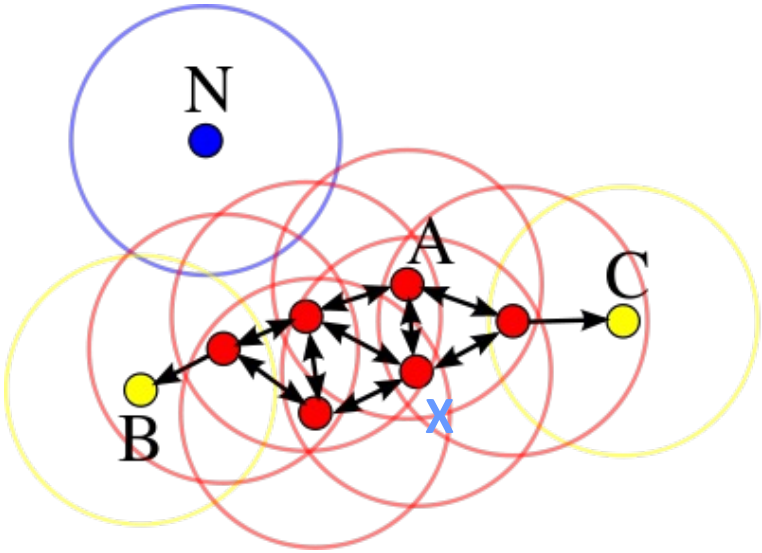
$$N(d(A, X_i) \leq \varepsilon) \geq n_{min}$$

- **Boundary points** are not surrounded by at least n_{min} points

- **Density Connected**: A point A is density-connected (reachable) to B if there's a sequence of directly reachable points between them:

$$A \rightarrow X_i \rightarrow X_{i+1} \rightarrow \ldots \rightarrow X_j \rightarrow \mathrm{B}$$

- **Noise**: Points that are neither core nor boundary.



Adapted from
https://en.wikipedia.org/wiki/DBSCAN

Adapted from
https://learning.oreilly.com/library/view/machine-learning-algorithms/9781789347999/50efb27d-abbe-4855-ad81-a5357050161f.xhtml

Density-based spatial clustering of applications with noise (DBSCAN)

- **Directly Reachable**: Point X is directly-reachable from point A if it is within a radius ε to A:

$$d(A, X) \leq \varepsilon$$

- **Core points:** Point A becomes a core point if it is surrounded by at least n_{min} points:

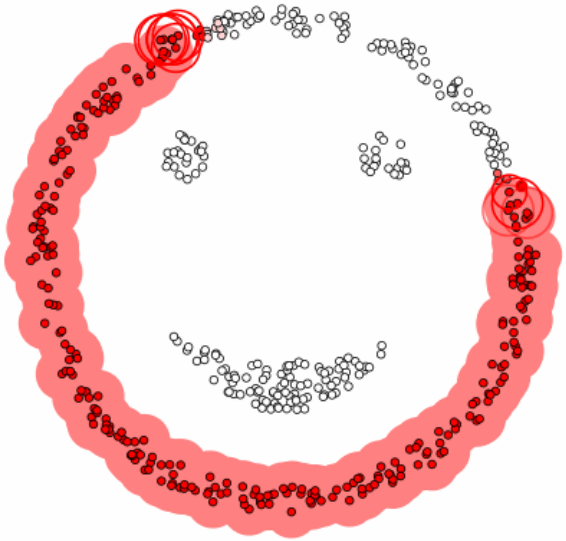
$$N(d(A, X_i) \leq \varepsilon) \geq n_{min}$$

- **Boundary points** are not surrounded by at least n_{min} points

- **Density Connected**: A point A is density-connected (reachable) to B if there's a sequence of directly reachable points between them:

$$A \rightarrow X_i \rightarrow X_{i+1} \rightarrow \ldots \rightarrow X_j \rightarrow \mathrm{B}$$

- **Noise**: Points that are neither core nor boundary.



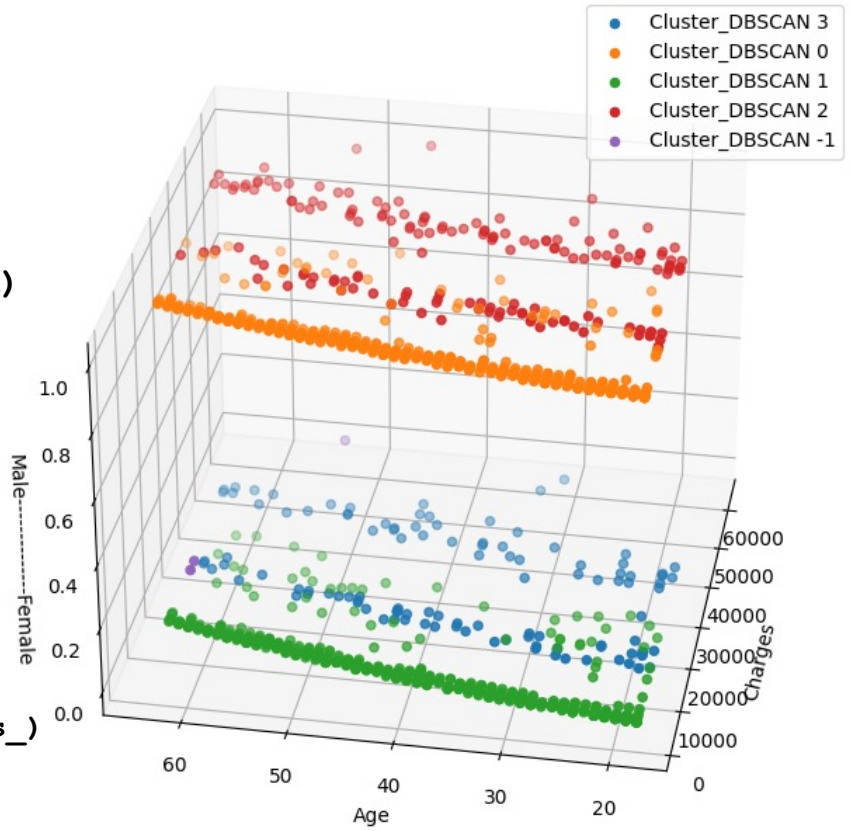
https://medium.com/analytics-vidhya/cluster-analysis-with-dbscan-density-based-spatial-clustering-of-applications-with-noise-6ade1ec23555

Adapted from
https://learning.oreilly.com/library/view/machine-learning-algorithms/9781789347999/50efb27d-abbe-4855-ad81-a5357050161f.xhtml

DBSCAN: Case Two - Health Insurance Cost Clustering

```
from sklearn.cluster import DBSCAN

# Apply DBSCAN
dbscaner = DBSCAN(eps=0.5, min_samples=80)
records['Cluster_DBSCAN'] = dbscaner.fit_predict(X_scaled)
```

Cluster -1 consists of 'noise'

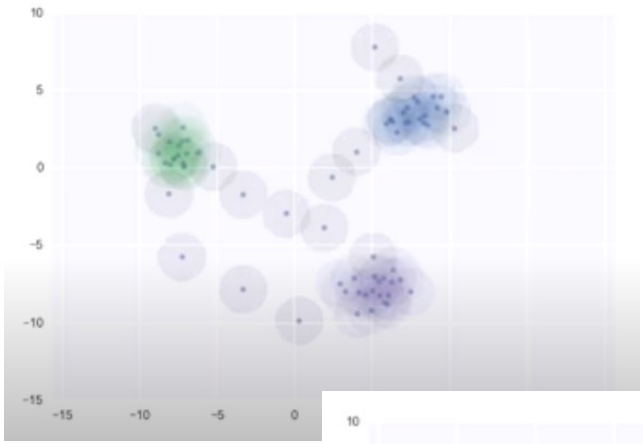


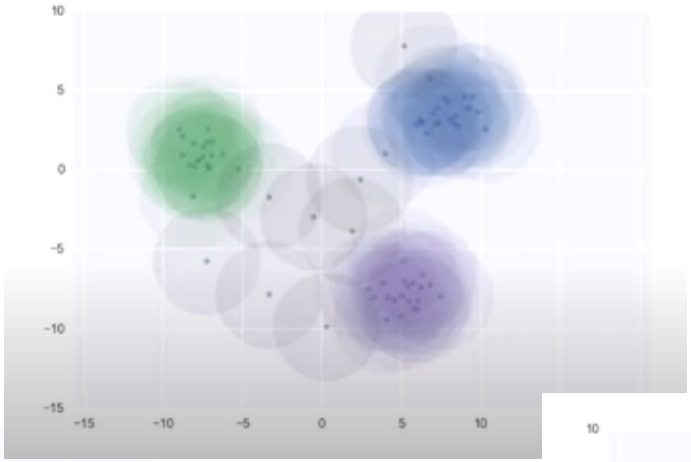
- Density-Based Clustering Validation (DBCV)
- Noise ratio

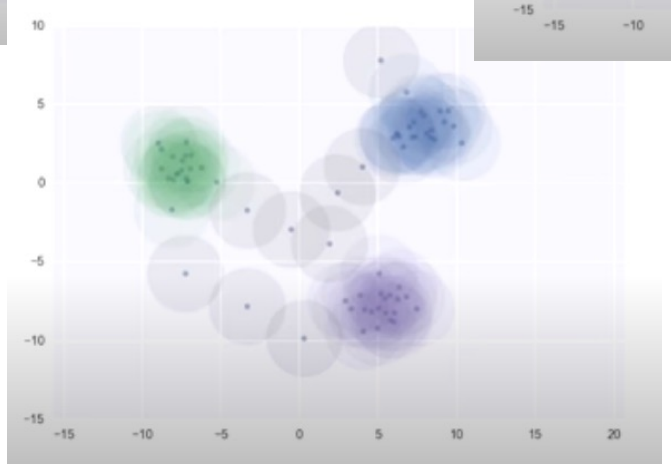
```
# Compute the noise ratio
noise_ratio = (dbscaner.labels_ == -1).sum() / len(dbscaner.labels_)
print(f"Noise Ratio: {noise_ratio}")
```

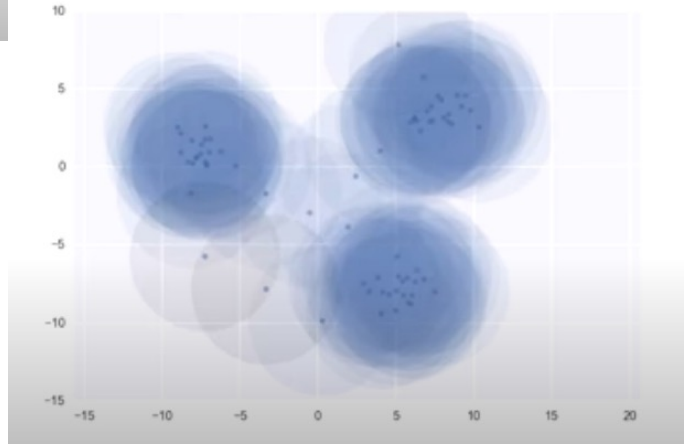
```
Noise Ratio: 0.0022421524663677130
```

HDBSCAN: Intuition







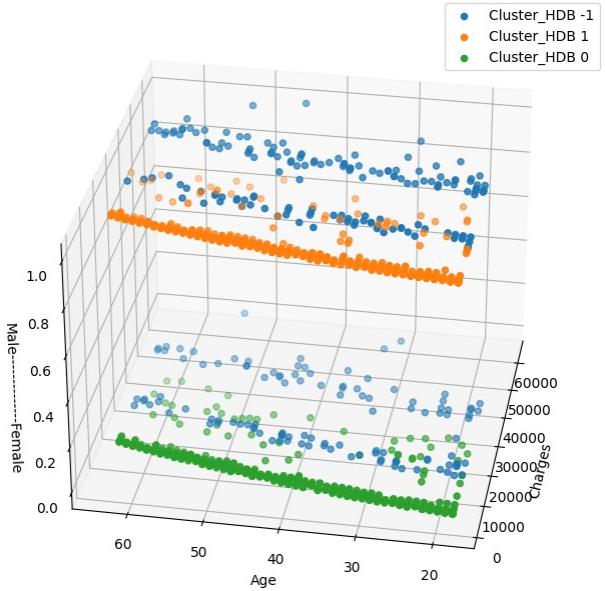


https://www.youtube.com/watch?v=dGsxd67IFiU&t=1384s

HDBSCAN: Case Two - Health Insurance Cost Clustering

An extension of the DBSCAN algorithm, by incorporating hierarchy, by varying density thresholds and detecting stable clusters.

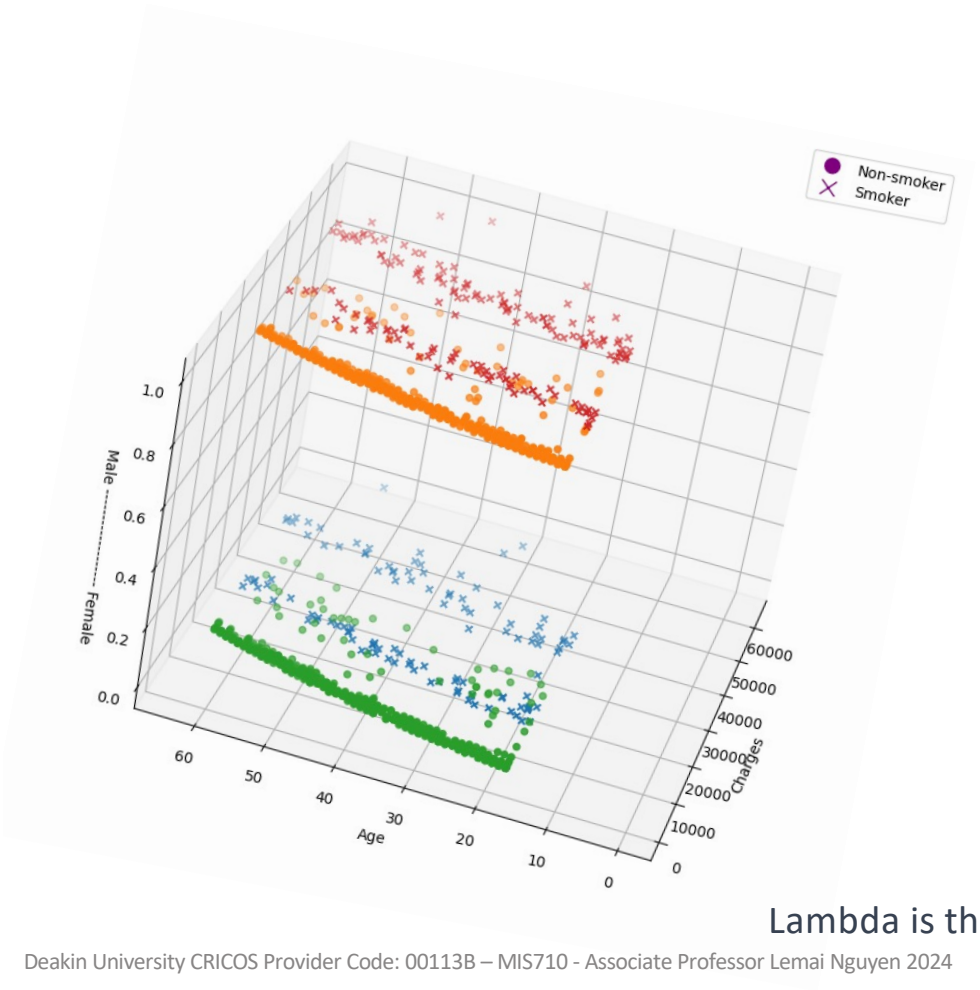
```
#Install HDBSCAN package
pip install hdbscan

import hdbscan

# Compute the clusters using HDBSCAN
hdbscaner = hdbscan.HDBSCAN(min_cluster_size=200)
hdbscaner.fit(X_scaled)

records['Cluster_HDB'] = hdbscaner.labels_


clusterer.condensed_tree_.plot()
```



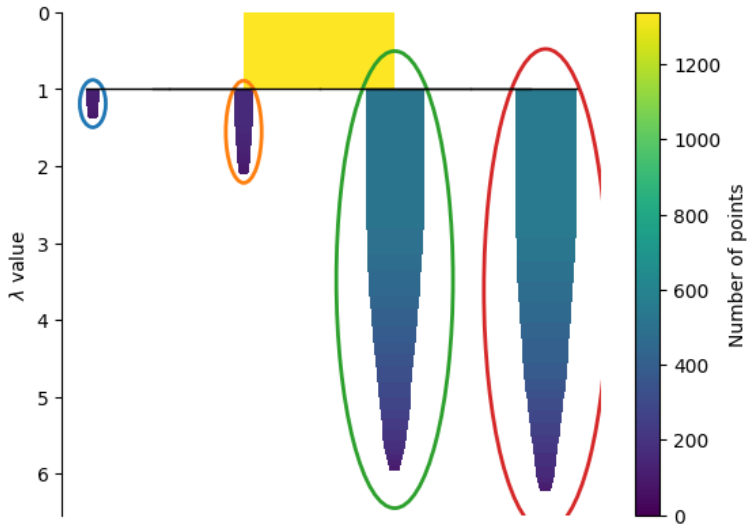
The negative cluster is noise

```
Noise Ratio: 0.20478325859491778
```

HDBSCAN: Case Two - Health Insurance Cost Clustering

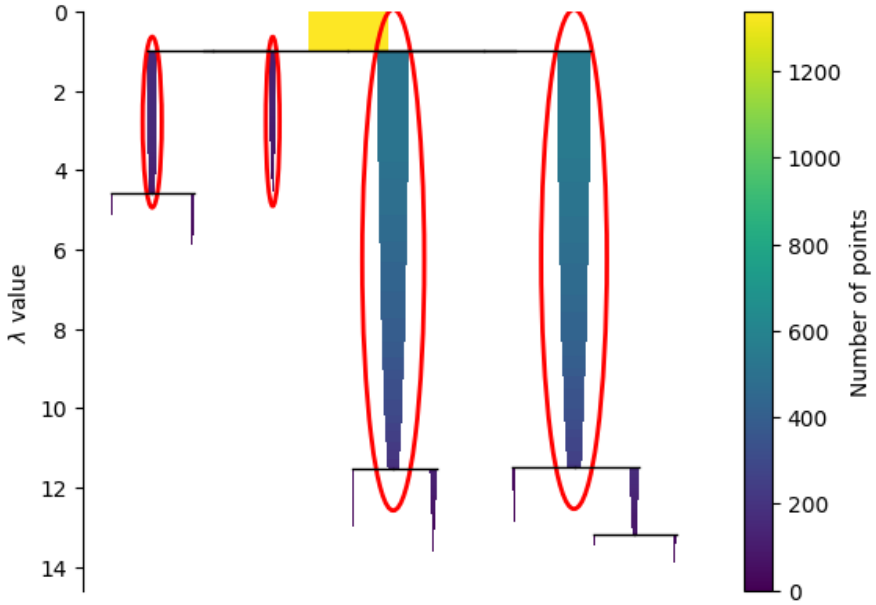



```
# Compute the clustering using HDBSCAN
hdbscaner = hdbscan.HDBSCAN(min_cluster_size=80)
hdbscaner.fit(X_scaled)

records['Cluster_HDB'] = hdbscaner.labels_

hdbscaner.condensed_tree_.plot(select_clusters=True)
```



Lambda is the inverse density level at which data points or clusters merge.

HDBSCAN: Case Two - Health Insurance Cost Clustering



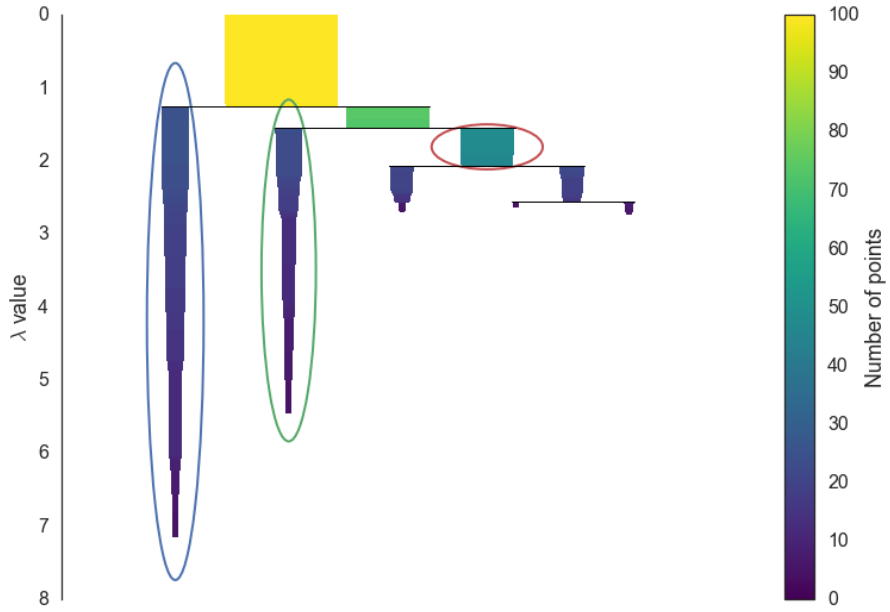

```
# Compute the clustering using HDBSCAN
hdbscaner = hdbscan.HDBSCAN(min_cluster_size=20)
hdbscaner.fit(X_scaled)

records['Cluster_HDB'] = hdbscaner.labels_

hdbscaner.condensed_tree_.plot(select_clusters=True)
```

Lambda is the inverse density level at which data points or clusters merge.

HDBSCAN: Hierarchical DBSCAN

An extension of the DBSCAN algorithm, flexible as it works with multiple scales

- Step 1: Convert data into a hierarchical structure
  - Create a minimum spanning tree based on data point distances
  - Build a dendrogram to represent hierarchical relationships
- Step 2: Identify clusters based on density
  - Define a density threshold to separate clusters
  - Prune branches in the dendrogram below the density threshold
- Step 3: Extract meaningful clusters and filter out noise
  - Retain clusters with a sufficient number of data points
  - Treat remaining data points as noise, not belonging to any cluster



https://hdbscan.readthedocs.io/en/latest/how_hdbscan_works.html

HDBSCAN: Business Applications

- Flexible and robust: Handles a wide range of data types and shapes, such as numerical, categorical, and mixed data
- Better insights: Discovers complex patterns in data that other algorithms might miss, including nested and irregularly shaped clusters
- Noise handling: Filters out irrelevant data points for cleaner results, allowing for more accurate decision-making
- Real-world applications:
  - Customer segmentation, market trend analysis
  - Fraud detection
  - Supply chain optimisation