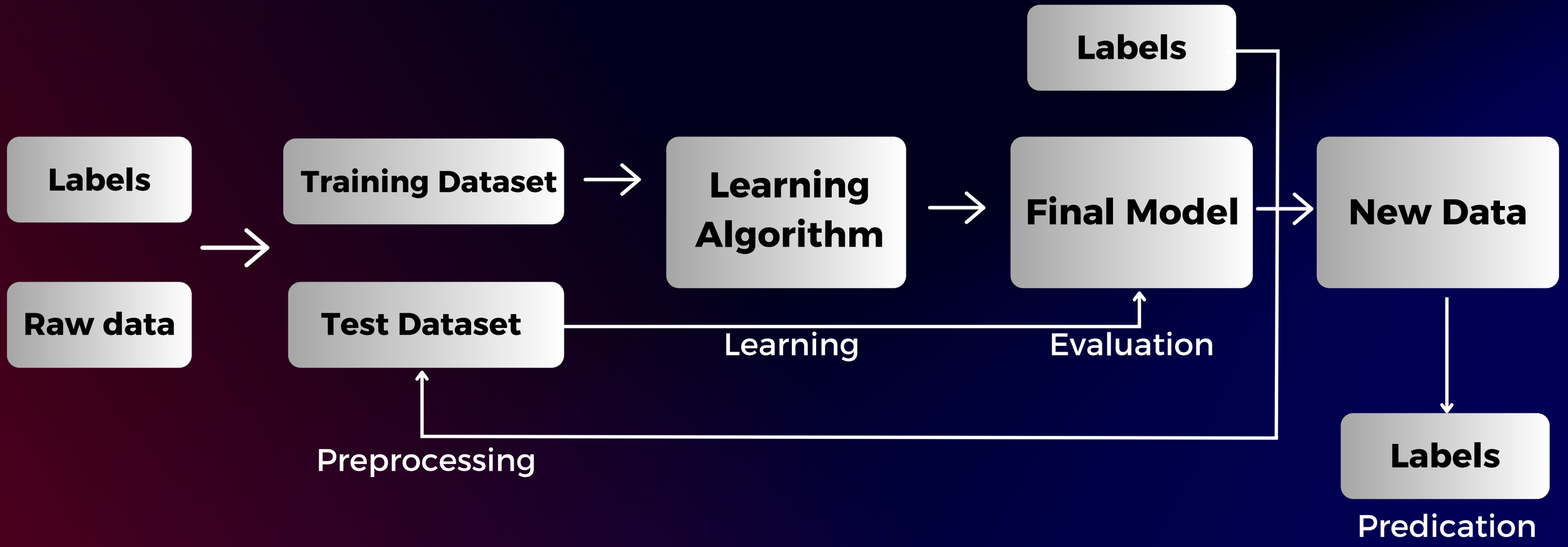
TOPIC: Clustering passengers on the Titanic

Machine Learning Workflow



Introduction to the K-means Algorithm

Idea:

- Input: Data and number of K clusters to find
- Output: Centers (M) and label vectors for each data point (Y)

Procedure:

1. Choose any K points as initial centers
2. Divide each data point into the cluster with the center closest to it.
3. If the assignment of data to each cluster in step 2 does not change compared to the previous loop, we stop the algorithm.
4. Update the center for each cluster by taking the average of all data points assigned to that cluster after step 2.
5. Go back to step 2.

Introduction to the K-means Algorithm

Limit:

- Need to know the number of clusters that need clustering
- The location of the cluster center will depend on the initial initialization point
- Not suitable for data sets with complex shapes or imbalances
- Sensitive to data with noisy values or unit magnitudes of variables

Application:

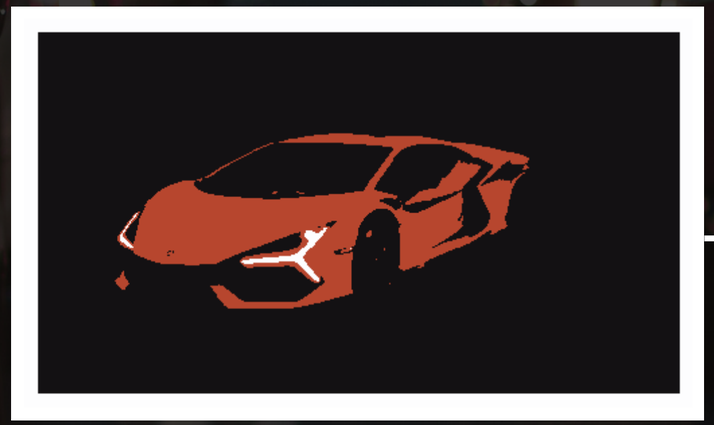
- Partition photos, videos, text, news,...
- Locate, identify, and group objects

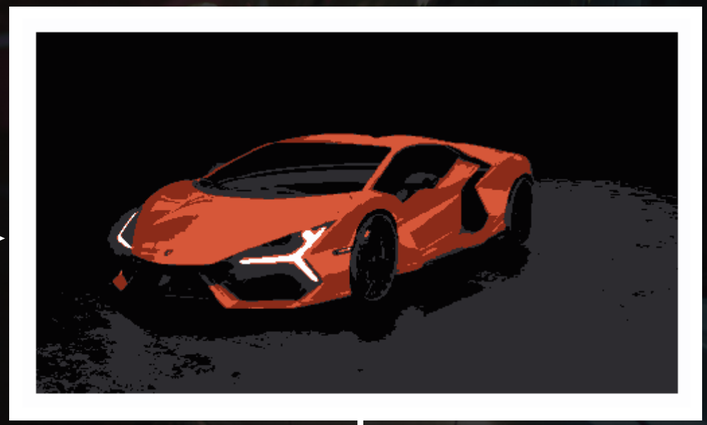
Introduction to the K-means Algorithm

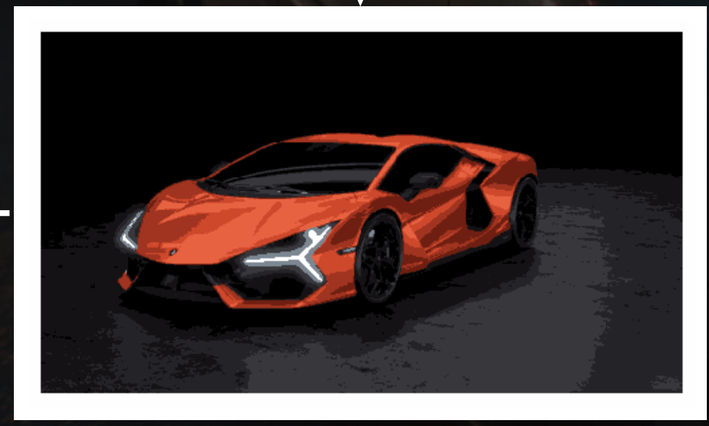
So how to determine the best number of clusters to divide for a specific data set?

- **Elbow**
- **Gap Statistics**
- **Silhouette**
- **Ch-index**

Application in Image Processing







Application in Image Processing

```
import numpy as np
import matplotlib.pyplot as plt
import requests

from PIL import Image
from io import BytesIO
from sklearn.cluster import KMeans
```

```
local_path = 'download.png'
img = Image.open(local_path)
img_array = np.array(img)

plt.imshow(img_array)
plt.axis('off')
plt.show()

X = img_array.reshape((img_array.shape[0]*img_array.shape[1], img_array.shape[2]))
```

```
for K in [3, 5, 10, 50]:
    kmeans = KMeans(n_clusters=K).fit(X)
    label = kmeans.predict(X)

    img4 = np.zeros_like(X)
    for k in range(K):
        img4[label == k] = kmeans.cluster_centers_[k]

    img5 = img4.reshape((img_array.shape[0], img_array.shape[1], img_array.shape[2]))
    plt.imshow(img5, interpolation='nearest')
    plt.axis('off')
    plt.show()
```

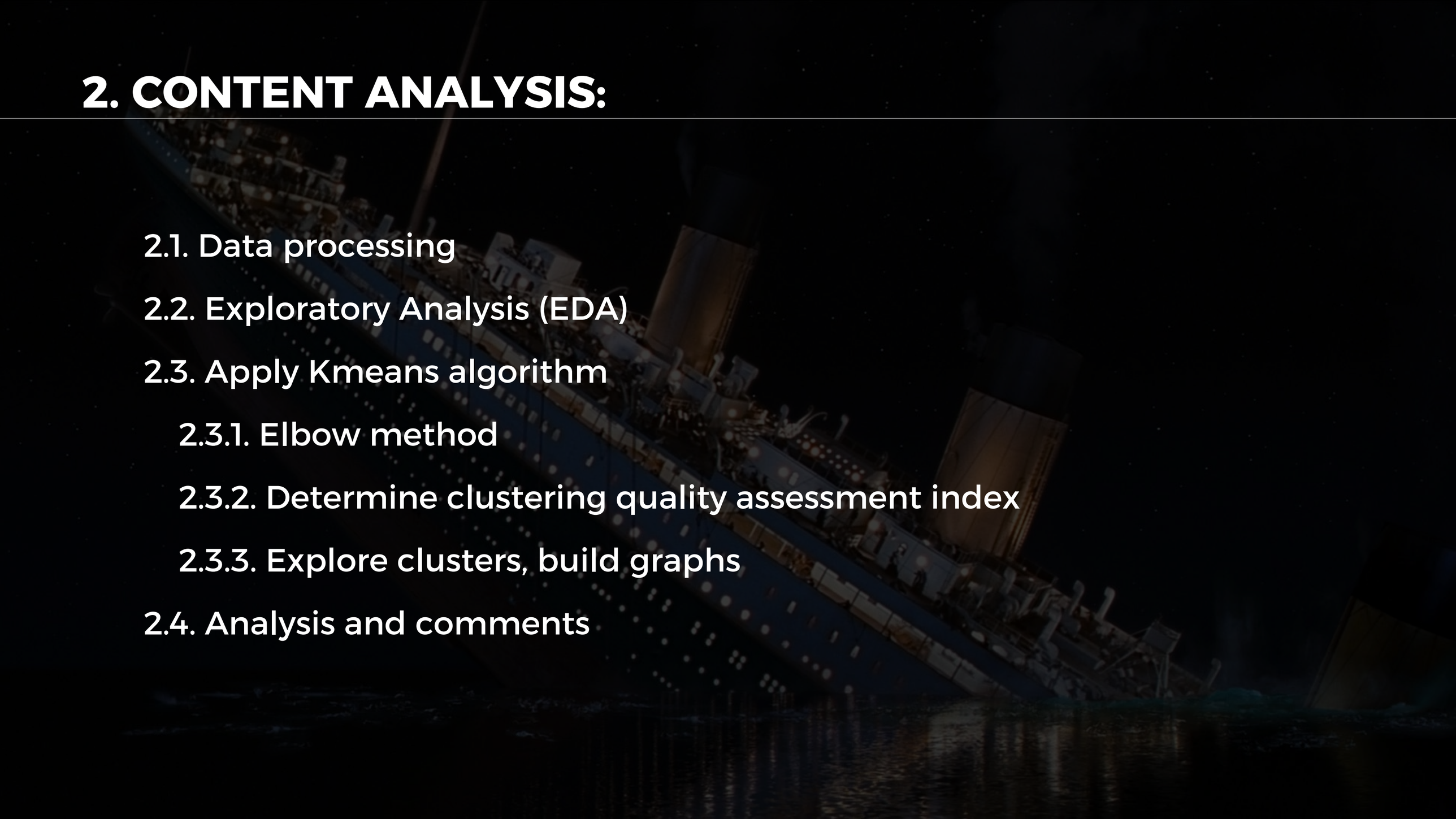
1. DATA DESCRIPTION:

Features:

Survival: Survive (float)
Pclass: Ticket class (int)
Name: Name (str)
Sex: Sex (str)
Age: Age (float)
Sibsp: Number of siblings on board (int)
Parch: Number of parents and children on board (int)
Ticket: Ticket number (str)
Fare: Fare (float)
Cabin: Cabin number (str)
Embarked: Port of embarkation (str)
Boat: Life vest (str)
Body: Body identification index (float)
Home.dest: Destination (str)

Link: https://bit.ly/4ah2qIn

2. CONTENT ANALYSIS:



2.1. Data processing

2.2. Exploratory Analysis (EDA)

2.3. Apply Kmeans algorithm

- 2.3.1. Elbow method
- 2.3.2. Determine clustering quality assessment index
- 2.3.3. Explore clusters, build graphs

2.4. Analysis and comments


THANK YOU !!!