Unsupervised Learning - Real Life Examples

## 1. Customer Segmentation for Marketing

### Problem

A retail company wants to segment its customers to run personalized marketing campaigns.

### Solution

Use K-Means clustering on features like purchase frequency, average order value, and recency.

### Implementation

```python  
from sklearn.cluster import KMeans  
model = KMeans(n\_clusters=4)  
model.fit(customer\_data)  
```

### Impact

The company tailored campaigns to each segment, improving conversion rates by 25%.

### Conclusion

Clustering allowed the business to uncover natural groupings without needing labeled data.

## 2. Anomaly Detection in Credit Card Transactions

### Problem

A bank wants to detect suspicious (fraudulent) transactions without prior examples of fraud.

### Solution

Use DBSCAN to detect outliers in transaction patterns.

### Implementation

```python  
from sklearn.cluster import DBSCAN  
db = DBSCAN(eps=0.5, min\_samples=5).fit(transactions)  
```

### Impact

Enabled real-time flagging of anomalous behavior with a 15% reduction in false positives.

### Conclusion

Unsupervised learning identified hidden fraud patterns with minimal labeled examples.

## 3. Document Clustering for News Categorization

### Problem

A news aggregator wants to group articles by topic without predefined categories.

### Solution

Use TF-IDF vectorization followed by KMeans clustering.

### Implementation

```python  
from sklearn.feature\_extraction.text import TfidfVectorizer  
vectorizer = TfidfVectorizer()  
X = vectorizer.fit\_transform(news\_texts)  
```

### Impact

Improved topic-based navigation and user personalization.

### Conclusion

Text clustering enhanced content discoverability and reading experience.

## 4. Product Recommendations via Association Rules

### Problem

An e-commerce site wants to suggest items frequently bought together.

### Solution

Use Apriori algorithm to find frequent itemsets.

### Implementation

```python  
from mlxtend.frequent\_patterns import apriori  
frequent\_itemsets = apriori(basket\_data, min\_support=0.1, use\_colnames=True)  
```

### Impact

Recommended combos increased cart value by 20%.

### Conclusion

Unsupervised rules mining revealed relationships without explicit labeling.

## 5. Image Compression using PCA

### Problem

Reduce storage cost by compressing high-resolution image datasets.

### Solution

Apply PCA to reduce the number of dimensions in pixel data.

### Implementation

```python  
from sklearn.decomposition import PCA  
pca = PCA(n\_components=50)  
compressed = pca.fit\_transform(images)  
```

### Impact

Achieved ~80% reduction in storage with minimal quality loss.

### Conclusion

Dimensionality reduction helped manage large visual datasets efficiently.