



UNIVERSITY OF TEHRAN

MACHINE LEARNING FINAL PROJECT

Classifying and Clustering images based on their authenticity (real or artificial)

Fardin Abbasi 810199456

Mansour Davoudi 810199567

Arash Mohamadpour 810197636

Amirhossein Safari 810101213

School of Electrical and Computer Engineering

Summer 2023

Table of Contents

1	Introduction	1
2	Data Preparation	2
2.1	Feature Extraction	2
2.2	Preprocessing	2
2.3	Dimension Reduction	4
3	Classification	6
3.1	Logistic Regression	6
3.1.1	Evaluation with Deep Features	7
3.1.2	Evaluation with Handcraft Features	8
3.2	SVM	9
3.2.1	Evaluation with Deep Features	10
3.2.2	Evaluation with Handcraft Features	11
3.3	Random Forest	12
3.3.1	Evaluation with Deep Features	12
4	Clustering	15
4.1	Mini Batch K-Means	16
4.1.1	Evaluation with Deep Features	17
4.1.2	Evaluation with Handcraft Features	21
4.2	Gaussian Mixture Models	24
4.2.1	Evaluation with Deep Features	25
4.2.2	Evaluation with Handcraft Features	26

Chapter 1

Introduction

In recent years, there has been a great focus on generating fake images using artificial intelligence algorithms. One of the well-known algorithms is the Generative Adversarial Network (GAN). A generative adversarial network is a class of machine learning frameworks in which two neural networks compete with each other in a zero-sum game, where one agent's gain is another agent's loss.

The final goal of this project is to classify fake images from real ones. Like a real-world project, we need to collect the raw dataset and then perform data cleansing and preprocessing. Before diving into the code, it is important to analyze the dataset to gain a good understanding of its behavior.

Now, let's take a look at how fake image generators like Dall-E and stable diffusion work. These types of generative AI are powered by a computer program called a diffusion model. In simple terms, a diffusion model destroys and recreates images to find statistical patterns within them.

Empirical results demonstrate that fake images generated by various models can be distinguished from real ones, as there exists a common artifact shared by fake images from different models. Fake photos can be effectively attributed to their source models, as different models leave unique fingerprints in their generated images.[1]

At first glance, the project can be divided into 4 parts:

1. Data Collection
2. Data Preparation
 - (a) Feature Extraction
 - (b) Preprocessing
 - (c) Dimension Reduction
3. Classifying
4. Clustering

Chapter 2

Data Preparation

2.1 Feature Extraction

In this project, besides the given deep features that are likely extracted from a CNN, we need to extract handcrafted features. Handcrafted features are features that are manually engineered by the data scientist.

The approach involved using two common techniques, namely Local Binary Patterns (LBP) and Fast Fourier Transform (FFT).

LBP is a texture descriptor technique that characterizes the local structure of an image by comparing the intensity of a central pixel with its surrounding neighbors. By applying LBP, the project aimed to capture relevant textural details that could contribute to the understanding and classification of the images or data.

FFT transforms a signal from the time domain to the frequency domain, enabling the identification of different frequency components within the signal. By using FFT, we extracted frequency-based features that could provide insights into the underlying patterns or characteristics of the data.

After applying LBP and FFT individually, the next step involved concatenating the extracted features. This process combined the information captured by both techniques, creating a more comprehensive feature representation. Subsequently, PCA was applied to reduce the dimensionality of the concatenated features. PCA aimed to identify the most significant patterns and reduce the feature space, allowing for efficient data analysis and potential improvement in subsequent tasks such as classification or clustering.

2.2 Preprocessing

Data preprocessing is an integral step in machine learning, as the quality of the data and the useful information that can be derived from it directly affect the ability of our model to learn. Therefore, it is necessary to preprocess our data before feeding it into our model.

In this project, the dataset has been preprocessed as below:

- i. Handling Null Values: In any real-world dataset, there are always a few null values.
- ii. Data Cleansing: Data cleansing is the process of identifying and correcting corrupt or inaccurate records in a dataset. It involves detecting incomplete, incorrect, inaccurate, or irrelevant parts of the data and then taking actions such as replacing, modifying, or deleting the problematic data. For instance:
 - a. There are a few incorrect labels, such as "forest" or "Jungle" instead of "jungle," "DALL.E," and other derivatives instead of "DALL-E."
 - b. Removing irrelevant parts of labels, including image formats and student IDs
- iii. Standardization: In Standardization, we transform our values such that the mean of the values is 0 and the standard deviation is 1.
- iv. Test, Train & Validation Split: Train data is used for training the model, validation data is used for tuning hyperparameters and choosing the best model, and test data is used for evaluating the chosen model.

	0	1	2	3	4	5	6	7	8	9	...	1270	1271	1272	1273	1274	1275
count	3.418000e+03	3.418000e+03	3.418000e+03	3.418000e+03	3.418000e+03	3.418000e+03	3.418000e+03	3.418000e+03	3.418000e+03	3.418000e+03	...	3.418000e+03	3.418000e+03	3.418000e+03	3.418000e+03	3.418000e+03	3.418000e+03
mean	1.039413e-17	3.326122e-17	2.078826e-17	3.949769e-17	5.456918e-17	3.326122e-17	3.637945e-18	5.404948e-17	4.573417e-17	4.313564e-17	...	1.143354e-17	1.455178e-17	5.716772e-17	1.767002e-17	3.326122e-17	1.039413e-17
std	1.000146e+00	1.000146e+00	1.000146e+00	1.000146e+00	1.000146e+00	1.000146e+00	1.000146e+00	1.000146e+00	1.000146e+00	1.000146e+00	...	1.000146e+00	1.000146e+00	1.000146e+00	1.000146e+00	1.000146e+00	1.000146e+00
min	-1.420355e+00	-7.216495e-01	-7.546811e-01	-7.755292e-01	-7.312001e-01	-8.199810e-01	-7.443181e-01	-1.028234e+00	-6.898393e-01	-5.702059e-01	...	-7.247044e-01	-7.508572e-01	-9.151319e-01	-5.520034e-01	-5.199426e-01	-7.901291e-01
25%	-7.655319e-01	-5.858331e-01	-5.784986e-01	-5.739029e-01	-5.749809e-01	-6.803629e-01	-5.667735e-01	-5.736277e-01	-5.684091e-01	-4.591309e-01	...	-5.870890e-01	-5.926085e-01	-5.589028e-01	-5.033773e-01	-3.840065e-01	-6.304769e-01
50%	-3.351053e-01	-4.545833e-01	-4.935805e-01	-4.631156e-01	-4.930848e-01	-5.053856e-01	-4.808223e-01	-2.839203e-01	-5.060833e-01	-4.124496e-01	...	-5.288289e-01	-5.06067e-01	-4.025278e-01	-4.613162e-01	-3.407365e-01	-5.435758e-01
75%	5.672952e-01	5.311868e-02	3.266890e-01	1.907713e-01	2.032570e-01	3.664433e-01	1.713867e-01	1.640915e-01	2.189193e-01	-1.139635e-01	...	2.946796e-01	2.380128e-01	1.552289e-01	7.862994e-02	-1.448412e-01	3.475486e-01
max	4.696961e+00	6.338370e+00	7.293078e+00	6.368939e+00	8.035126e+00	5.132912e+00	9.123699e+00	1.004787e+01	9.085969e+00	1.081767e+01	...	6.362443e+00	6.768572e+00	6.797745e+00	7.933971e+00	1.074608e+01	5.428087e+00

Figure 2.1

As you see, features are transformed such that the mean of each feature is 0 and the standard deviation is 1.

Labels Summary:

none	1708	mountain	1140	fake	1710
stable	804	sea	1140	real	1708
dalle	776	jungle	1138	Name: class, dtype: int64	
dream	70				
crayon	30				
midjourney	30				
Name: generator, dtype: int64		Name: category, dtype: int64			

(a)
(b)
(c)

Figure 2.2: Labels Summary

2.3 Dimension Reduction

Dimension reduction is a fundamental technique in data analysis and machine learning. Its goal is to reduce the number of variables or features in a dataset while preserving the essential information. This technique is especially valuable when working with high-dimensional data, as it helps address challenges related to visualization, computational efficiency, and interpretability.

Significance of Dimension Reduction:

1. **Simplification of Data:** High-dimensional datasets can be complex and difficult to analyze. Dimension reduction helps simplify the data representation by reducing the number of variables, making it easier to understand and interpret.
2. **Visualization:** Dimension reduction techniques allow us to visualize data in lower-dimensional spaces, such as 2D or 3D, facilitating a better understanding of patterns and relationships among the data points.
3. **Noise Reduction:** By eliminating irrelevant or redundant features, dimension reduction can help reduce the impact of noise or irrelevant data, improving the quality of subsequent analyses or machine learning models.
4. **Computational Efficiency:** High-dimensional data often requires significant computational resources. By reducing the dimensionality, the computational complexity can be reduced, resulting in faster processing and analysis.

In this project we use the PCA and LOL techniques to reduce dimension.

Linear Optimal Low-Rank Projection (LOL):

The key intuition behind LOL is that we can jointly use the means and variances from each class (like LDA and CCA), but without requiring more dimensions than samples (like PCA), or restrictive sparsity assumptions. Using random matrix theory, we are able to prove that when the data are sampled from a Gaussian, LOL finds a better low-dimensional representation than PCA, LDA, CCA, and other linear methods.

Principal Component Analysis (PCA):

PCA is a widely used technique for dimension reduction. It identifies a new set of variables, called principal components, that are linear combinations of the original features. These components are ordered in terms of the amount of variance they explain in the data.

Reduced-Dimensional Data with PCA Components

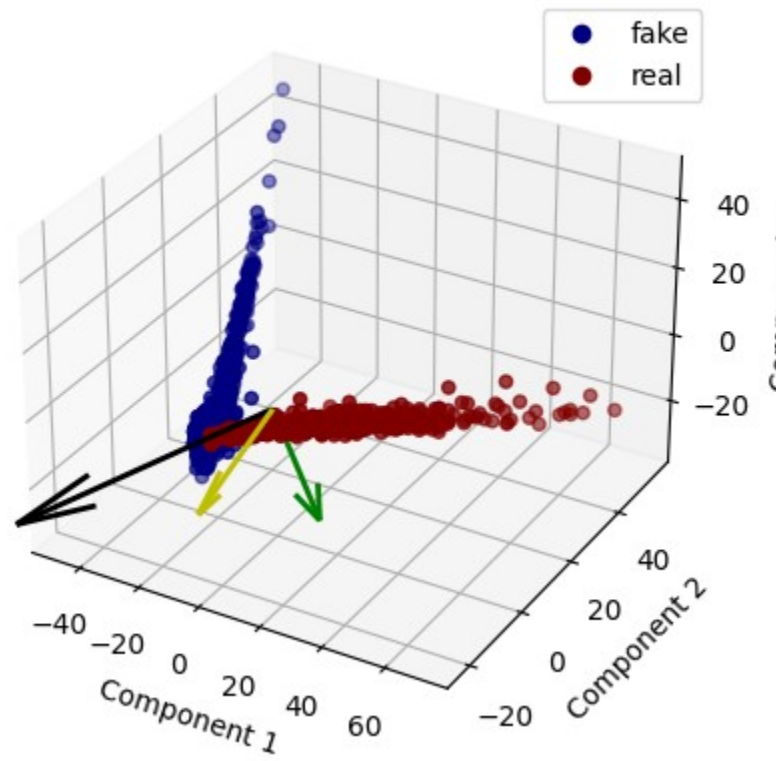


Figure 2.3

Chapter 3

Classification

Classification is a supervised machine learning method where the model aims to predict the correct label of a given input data. In classification, the model is trained using the training data and then evaluated on test data before being used to make predictions on new unseen data.

There are several methods for classification, and each is preferred based on different circumstances, such as the distribution and size of the dataset. In this project, we will test several popular methods to assess their performance.

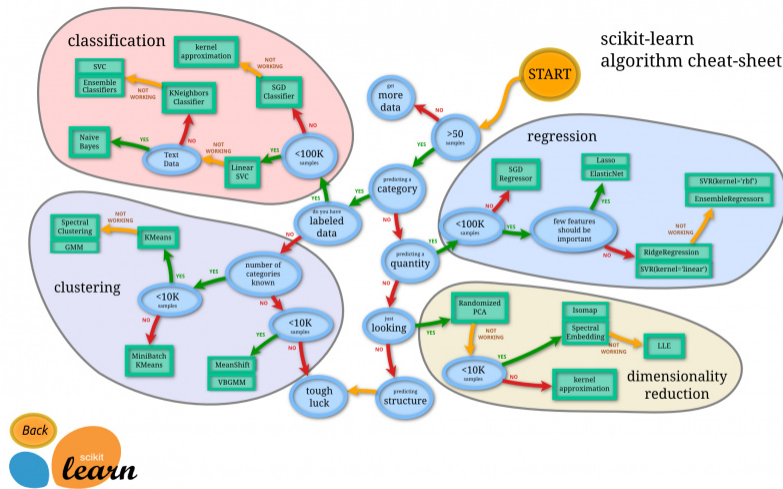


Figure 3.1: scikit-learn algorithm cheat sheet

3.1 Logistic Regression

Logistic regression is a classification algorithm used to determine the probability of event success and event failure. It is employed when the dependent variable is binary (0/1, True/False, Yes/No) in nature. Logistic regression categorizes data into discrete classes by analyzing the relationship within a labeled dataset. It learns a linear relationship from the provided dataset and incorporates non-linearity through the use of the Sigmoid function.[2]

Advantages	Disadvantages
Logistic regression is easier to implement, interpret, and very efficient to train.	If the number of observations is lesser than the number of features, Logistic Regression should not be used, otherwise, it may lead to overfitting.
It makes no assumptions about distributions of classes in feature space.	It constructs linear boundaries.
It can easily extend to multiple classes(multinomial regression) and a natural probabilistic view of class predictions.	The major limitation of Logistic Regression is the assumption of linearity between the dependent variable and the independent variables.
It not only provides a measure of how appropriate a predictor(coefficient size)is but also its direction of association (positive or negative).	It can only be used to predict discrete functions. Hence, the dependent variable of Logistic Regression is bound to the discrete number set.
It is very fast at classifying unknown records.	Non-linear problems can't be solved with logistic regression because it has a linear decision surface. Linearly separable data is rarely found in real-world scenarios.
Good accuracy for many simple data sets and it performs well when the dataset is linearly separable.	Logistic Regression requires average or no multicollinearity between independent variables.
It can interpret model coefficients as indicators of feature importance.	It is tough to obtain complex relationships using logistic regression. More powerful and compact algorithms such as Neural Networks can easily outperform this algorithm.
high-dimensional datasets. One may consider Regularization (L1 and L2) techniques to avoid over-fitting in these scenarios.	But Logistic Regression needs that independent variables are linearly related to the log odds ($\log(p/(1-p))$).
Logistic regression is less inclined to over-fitting but can overfit	In Linear Regression, independent and dependent variables are related linearly.

Table 3.1

3.1.1 Evaluation with Deep Features

For training the model, the "Newton-Cholesky" solver is used, which is recommended when the number of samples is much larger than the number of features.

The classification report and the confusion matrix are shown below, demonstrating the performance of the model:

	precision	recall	f1-score	support
fake	1.00	0.99	1.00	519
real	0.99	1.00	1.00	507
accuracy			1.00	1026
macro avg	1.00	1.00	1.00	1026
weighted avg	1.00	1.00	1.00	1026

Figure 3.2: classification report

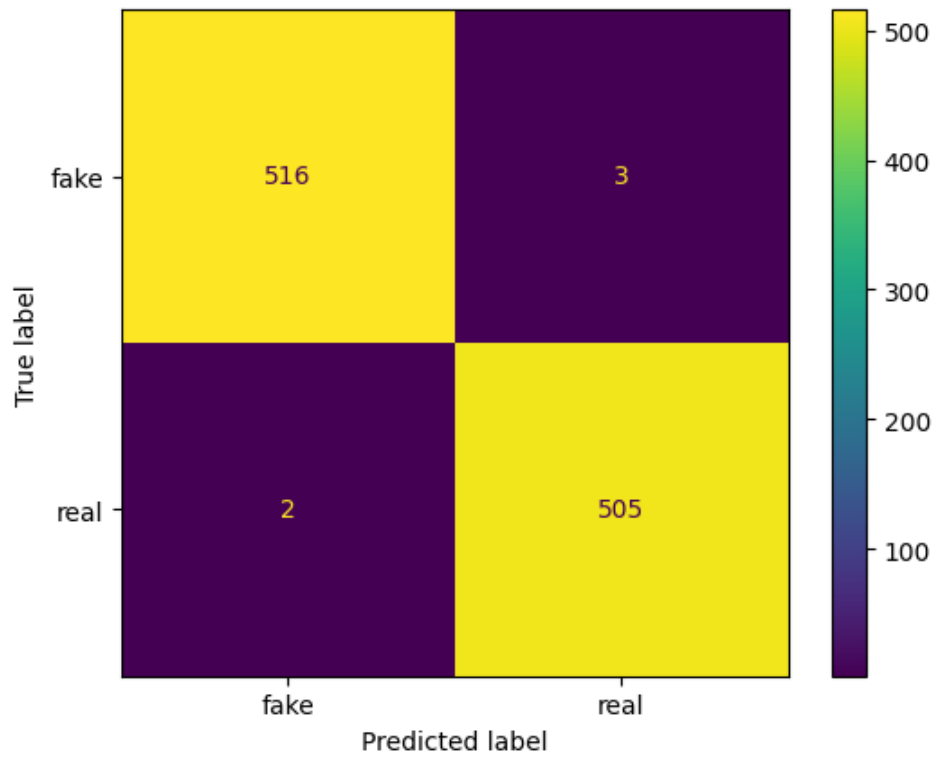


Figure 3.3: Confusion Matrix

3.1.2 Evaluation with Handcraft Features

The classification report and the confusion matrix, shown below, demonstrate that our extracted features do not classify well using Logistic Regression:

	precision	recall	f1-score	support
fake	0.69	0.71	0.70	490
real	0.73	0.71	0.72	536
accuracy			0.71	1026
macro avg	0.71	0.71	0.71	1026
weighted avg	0.71	0.71	0.71	1026

Figure 3.4: classification report

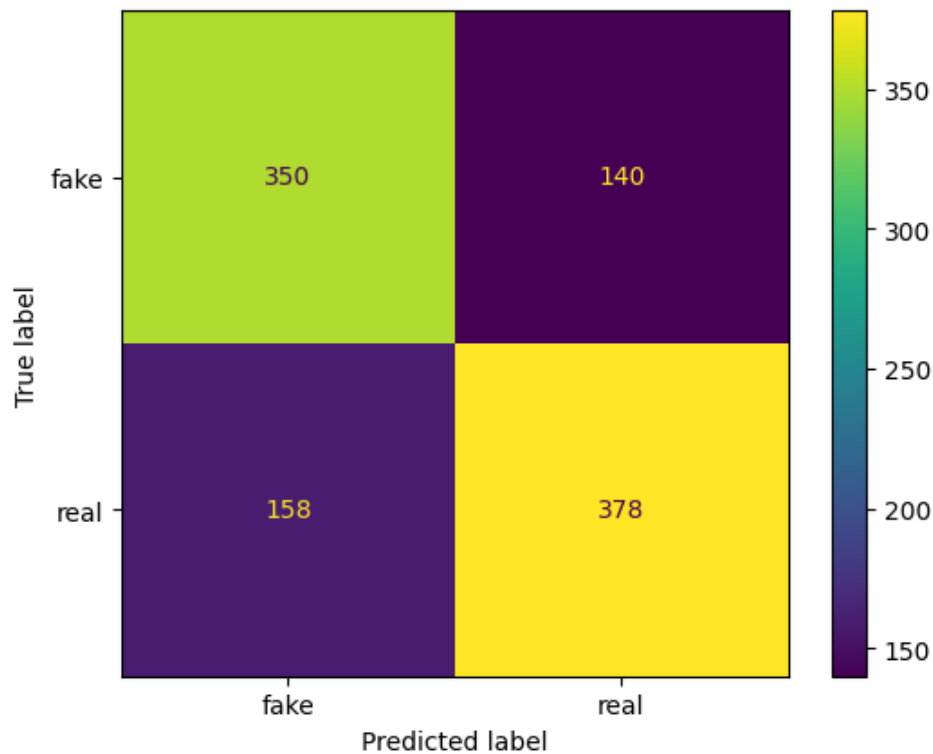


Figure 3.5: Confusion Matrix

3.2 SVM

SVM is a supervised machine learning algorithm that aids in classification or regression problems. Its goal is to find an optimal boundary between the possible outputs. In its simplest form, SVM is applied to binary classification, dividing data points into either 1 or 0.

Advantages:

- SVM works relatively well when there is a clear margin of separation between classes.
- SVM is more effective in high-dimensional spaces.
- SVM is effective in cases where the number of dimensions is greater than the number of samples.
- SVM is relatively memory efficient

Disadvantages:

- The SVM algorithm is not suitable for large data sets.
- SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.
- In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
- As the support vector classifier works by putting data points, above and below the classifying hyperplane there is no probabilistic explanation for the classification.[3]

3.2.1 Evaluation with Deep Features

$$\min_{w,b,\xi} \frac{1}{2} \|w\|^2 + C \sum_{i=1}^n \xi_i$$

subject to:

$$\begin{aligned} y_i(w^T x_i + b) &\geq 1 - \xi_i, \quad i = 1, 2, \dots, n \\ \xi_i &\geq 0, \quad i = 1, 2, \dots, n \end{aligned}$$

C is a parameter which determines the trade-off between lower error or higher margin. In order to choose this hyperparameter, we used grid search technique and the best C equals to 0.1.

The classification report and the confusion matrix are shown as below which demonstrate how well the model works:

	precision	recall	f1-score	support
fake	1.00	0.99	1.00	519
real	0.99	1.00	1.00	507
accuracy			1.00	1026
macro avg	1.00	1.00	1.00	1026
weighted avg	1.00	1.00	1.00	1026

Figure 3.6: classification report

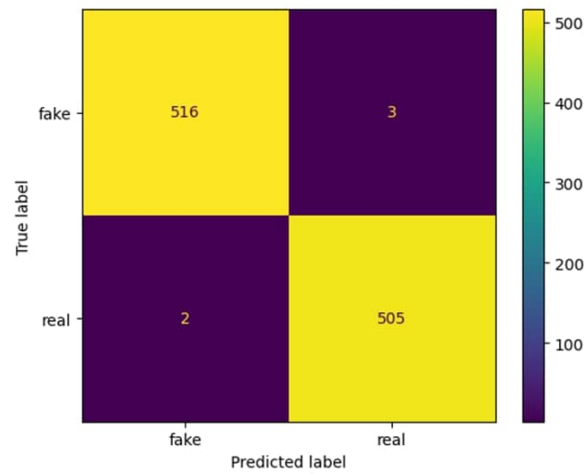


Figure 3.7: Confusion Matrix

3.2.2 Evaluation with Handcraft Features

The classification report and confusion matrix shown below demonstrate that this classification incorrectly classifies all the images as "fake" or "real," resulting in poor classification performance:

	precision	recall	f1-score	support
fake	0.69	0.72	0.70	490
real	0.73	0.70	0.71	536
accuracy			0.71	1026
macro avg	0.71	0.71	0.71	1026
weighted avg	0.71	0.71	0.71	1026

Figure 3.8: classification report

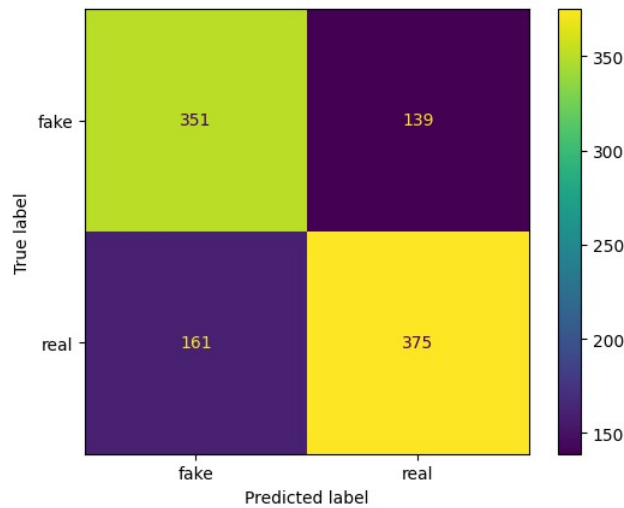


Figure 3.9: Confusion Matrix

3.3 Random Forest

Random forests or random decision forests are an **ensemble learning** method for classification, regression, and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees. For regression tasks, the mean or average prediction of the individual trees is returned.[4]

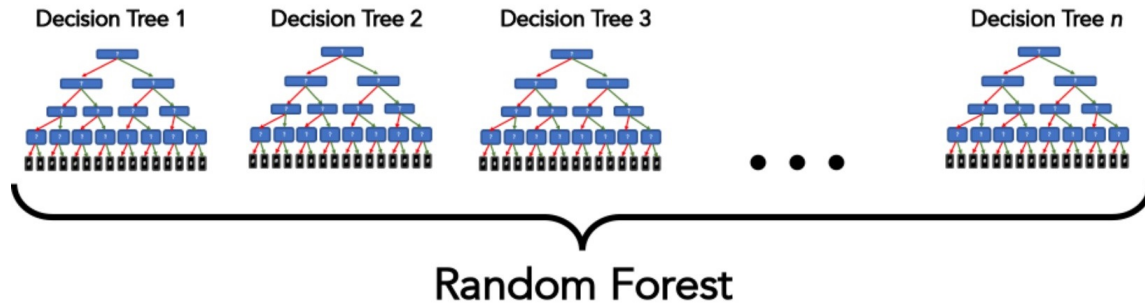


Figure 3.10: Random Forest

Random Forest Features:

- The accuracy of Random forest is generally very high.
- Its efficiency is particularly Notable in Large Data sets.
- Provides an estimate of important variables in classification.
- Forests Generated can be saved and reused.
- Unlike other models, It doesn't overfit with more features.

3.3.1 Evaluation with Deep Features

Two important hyperparameters to find in random forest method, are the number of estimators and the maximum depth. The Best Hyperparameters are found by Randomized Search CV

Best Hyperparameters: {'n_estimators': 85, 'max_depth': 100}

The classification report and the confusion matrix are shown as below which demonstrate how well the model works:

	precision	recall	f1-score	support
fake	1.00	0.99	1.00	519
real	0.99	1.00	1.00	507
accuracy			1.00	1026
macro avg	1.00	1.00	1.00	1026
weighted avg	1.00	1.00	1.00	1026

Figure 3.11: classification report

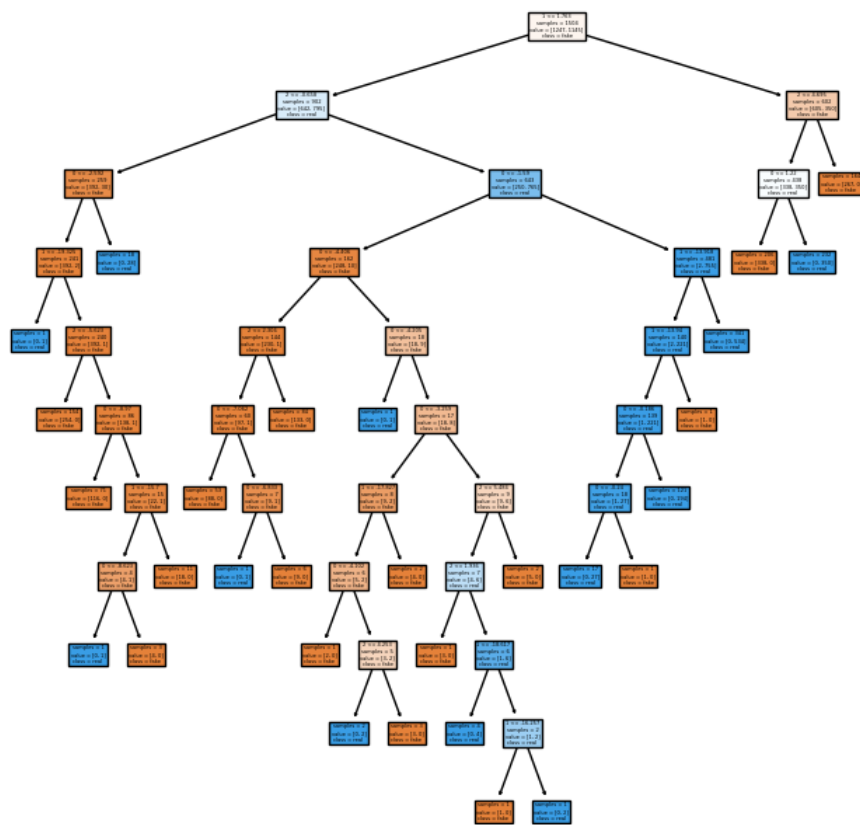


Figure 3.14

Chapter 4

Clustering

Clustering is the task of partitioning unlabeled data or data points into different clusters based on their similarities, with the objective of grouping similar data points together while distinguishing them from other groups. In other words, the clustering process aims to identify clusters or groups with similar characteristics and assign data points to these clusters.

All clustering methods can roughly be divided into four groups:

1. Hierarchical clustering
2. Centroid-based clustering
3. Graph-based clustering
4. Density-based clustering

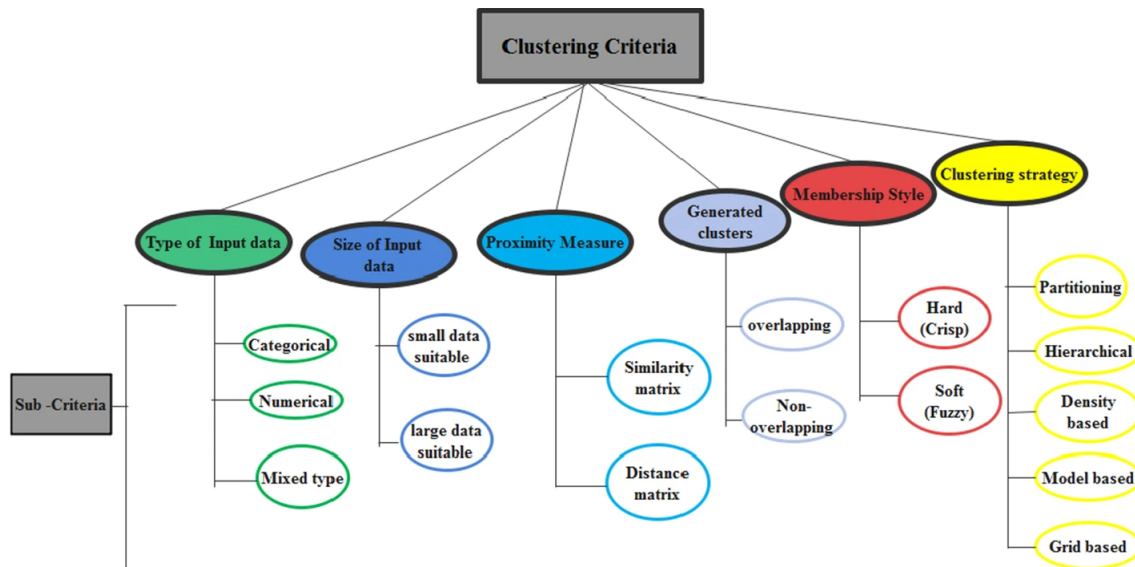


Figure 4.1: Clustering Criteria

4.1 Mini Batch K-Means

K-means is one of the most popular clustering algorithms, primarily due to its efficient time performance. However, as the size of the datasets being analyzed increases, the computation time of K-means also increases because it requires the entire dataset to be stored in memory. To address this issue, several methods have been proposed to reduce the computational and memory costs of the algorithm. One such approach is the **Mini-Batch K-means algorithm**.

The Mini-Batch K-means algorithm is faster than the traditional batch K-means algorithm but may yield slightly different results due to the use of mini-batches for computations.[5]

Pros:

1. Simple: It is easy to implement k-means and identify unknown groups of data from complex data sets. The results are presented easily and simply.
2. Flexible: K-means algorithm can easily adjust to the changes. If there are any problems, adjusting the cluster segment will allow changes to easily occur in the algorithm.
3. Suitable in a large dataset: K-means is suitable for a large number of datasets and it's computed much faster than the smaller dataset. It can also produce higher clusters.
4. Efficient: The algorithm used is good at segmenting the large data set. Its efficiency depends on the shape of the clusters. K-means works well in hyper-spherical clusters.
5. Time complexity: K-means segmentation is linear in the number of data objects thus increasing execution time. It doesn't take more time in classifying similar characteristics in data like hierarchical algorithms.
6. Computation cost: Compared to using other clustering methods, a k-means clustering technique is fast and efficient in terms of its computational cost $O(K*n*d)$.
7. Accuracy: K-means analysis improves clustering accuracy and ensures information about a particular problem domain is available. Modification of the k-means algorithm based on this information improves the accuracy of the clusters.
8. Spherical clusters: This mode of clustering works great when dealing with spherical clusters. It operates with an assumption of joint distributions of features since each cluster is spherical. All the clusters' features or characters have equal variance and each is independent of the other.

Cons:

1. No-optimal set of clusters: K-means doesn't allow the development of an optimal set of clusters and for effective results, you should decide on the clusters before.
2. Lacks consistency: K-means clustering gives varying results on different runs of an algorithm. A random choice of cluster patterns yields different clustering results resulting in inconsistency.
3. Uniform effect: It produces clusters with uniform sizes even when the input data has different sizes.
4. Order of values: How data is ordered in building the algorithm affects the final results of the data set.
5. Handle numerical data: K-means algorithm can be performed in numerical data only.
6. Operates in assumption: K-means clustering technique assumes that we deal with spherical clusters and each cluster has equal numbers for observations. The spherical assumptions have to be satisfied. The algorithm can't work with clusters of unusual size.
7. Specify K-values: For K-means clustering to be effective, you have to specify the number of clusters (K) at the beginning of the algorithm.
8. Prediction issues: Predicting the k-values or the number of clusters is difficult. It is also difficult to compare the quality of the produced clusters.[6]

Inertia is the sum of the squared distances of samples to their closest cluster center. The goal is to minimize this value. However, if we choose K to be equal to the number of samples, the inertia will be 0. While this is the smallest possible inertia value, it does not achieve our goal of clustering the data into an optimal number of clusters.

As the number of clusters increases, the inertia value decreases. Therefore, we need to manually select K while considering the trade-off between the inertia value and the number of clusters. The Elbow Method is commonly used to determine the appropriate value of K. We plot the inertia values against the number of clusters and choose the "elbow point" on the graph. After this point, the improvement in the inertia value becomes less significant, indicating that adding more clusters may not provide significant benefits.

4.1.1 Evaluation with Deep Features

The clustering results for different number of clusters are shown as below:

Mini Batch K-Means Clustering (2 clusters)

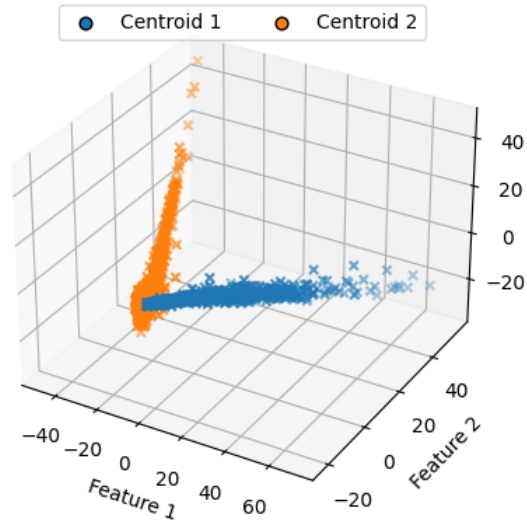


Figure 4.2

Mini Batch K-Means Clustering (3 clusters)

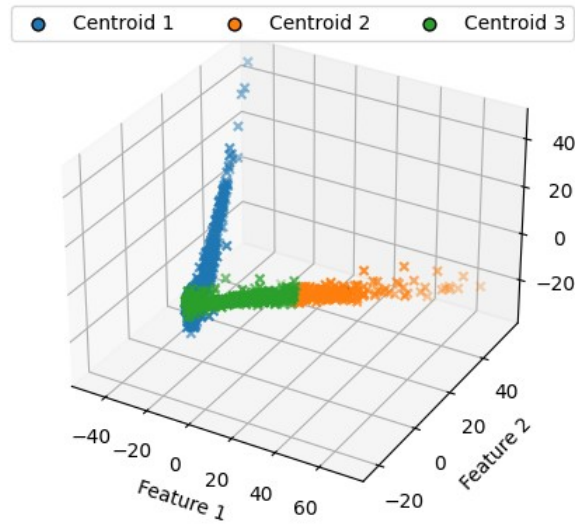


Figure 4.3

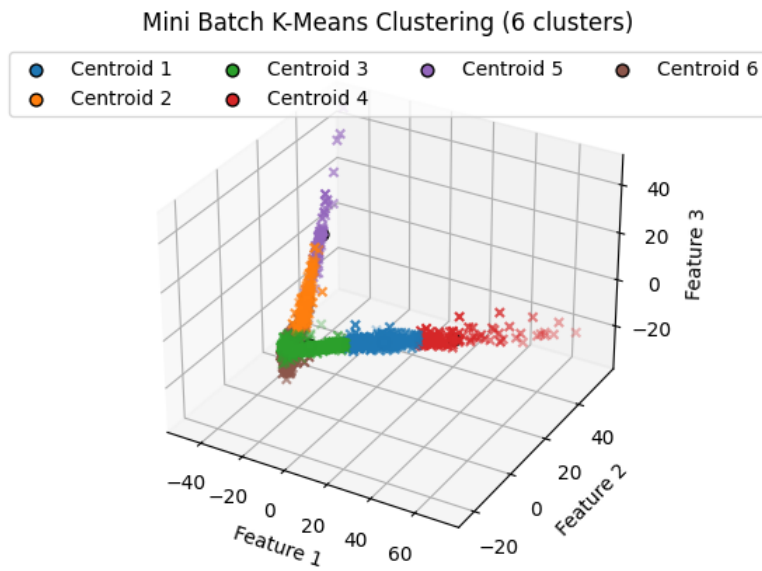


Figure 4.4

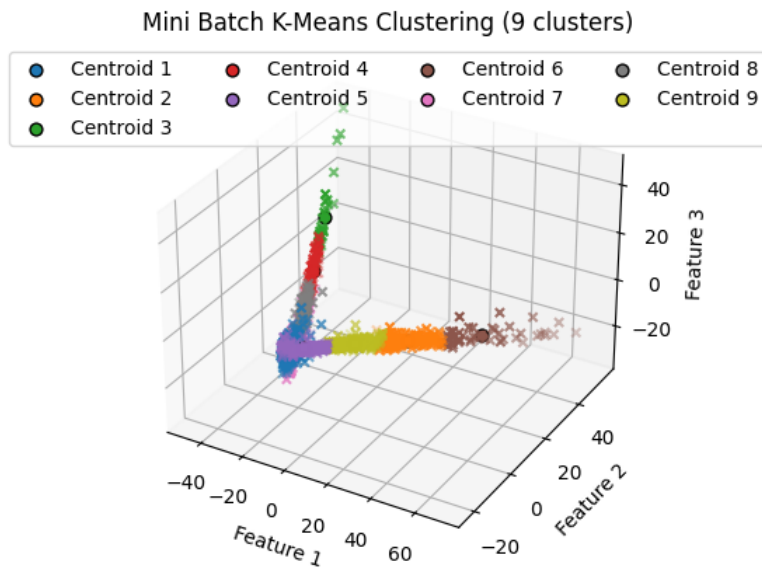


Figure 4.5

Mini Batch K-Means Clustering (50 clusters)

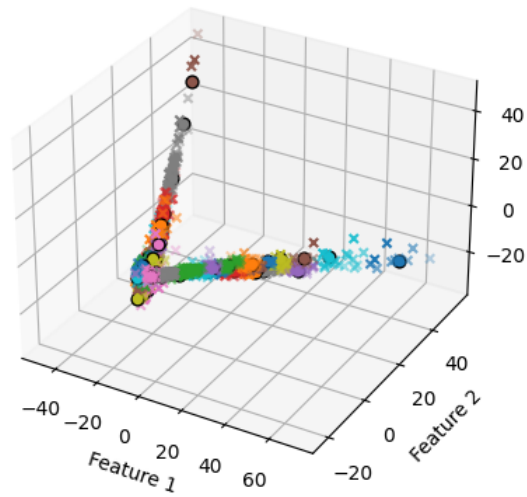


Figure 4.6

The best number of clusters, is elbow point in the plot of inertia with respect to number of clusters:

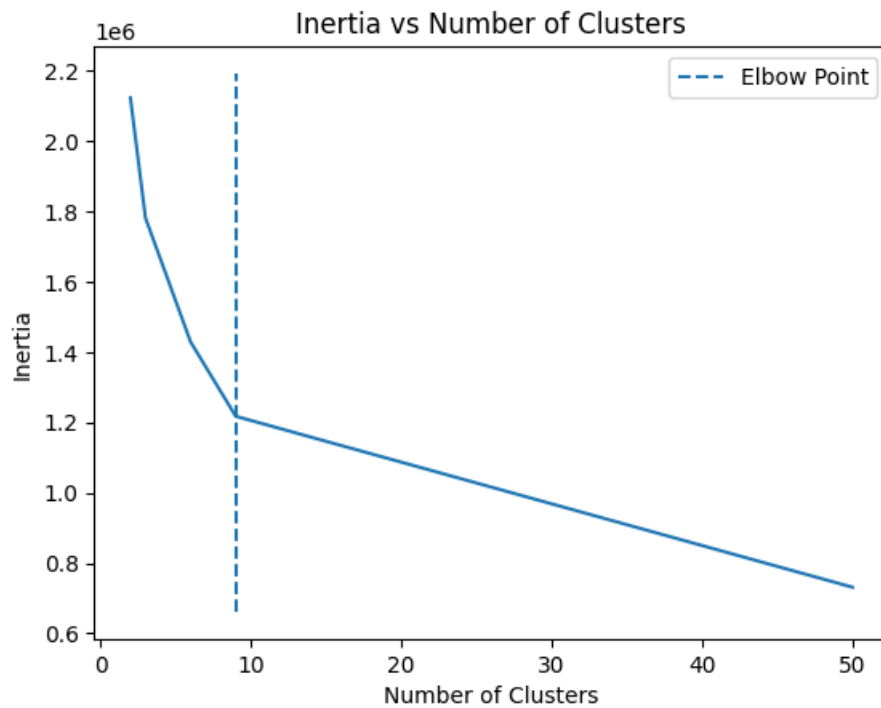


Figure 4.7

4.1.2 Evaluation with Handcraft Features

The clustering results for different number of clusters are shown as below:

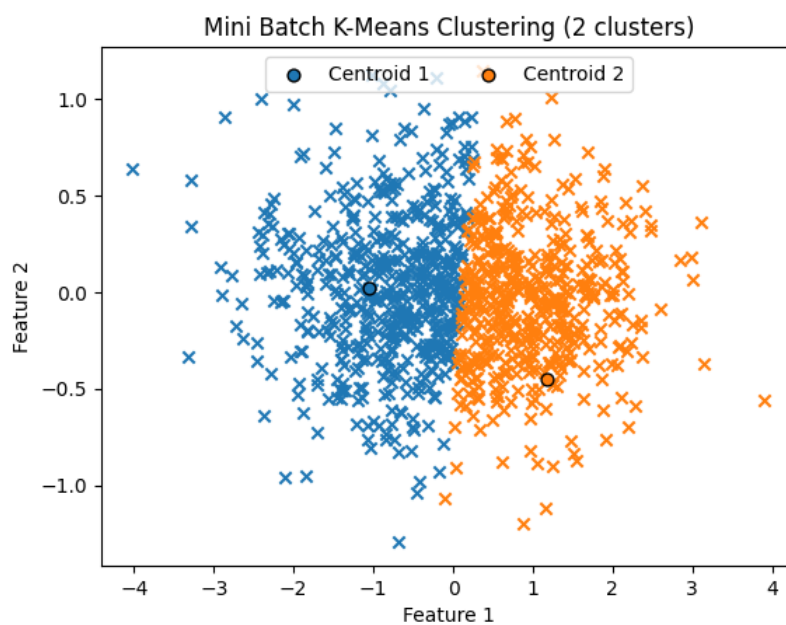


Figure 4.8

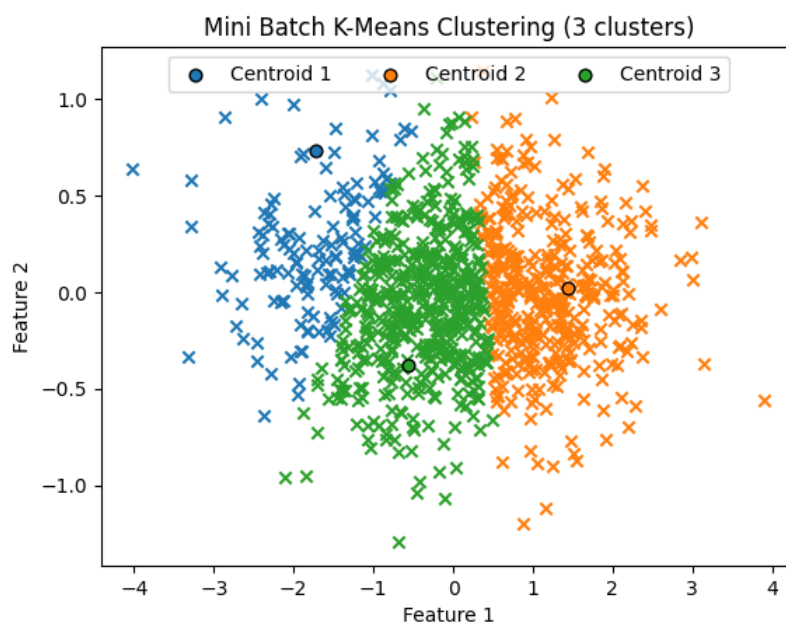


Figure 4.9

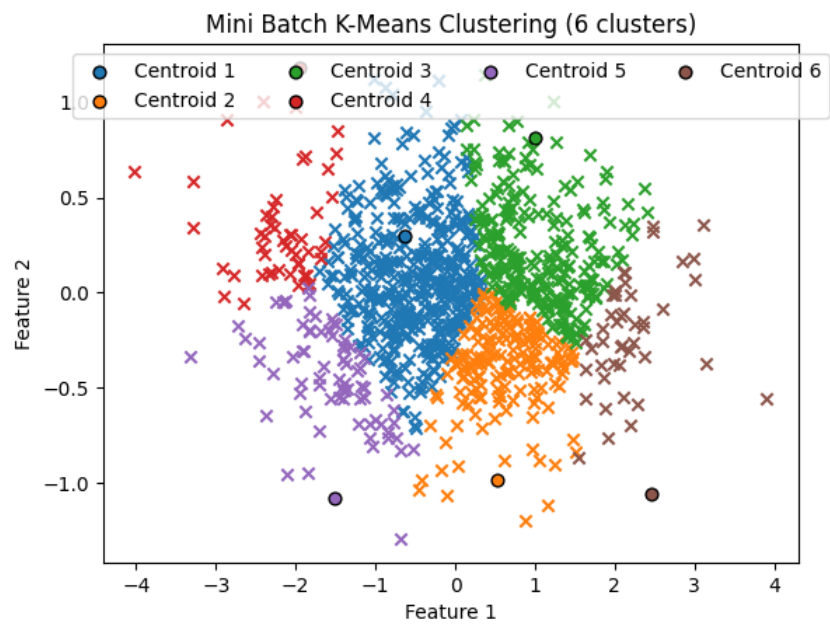


Figure 4.10

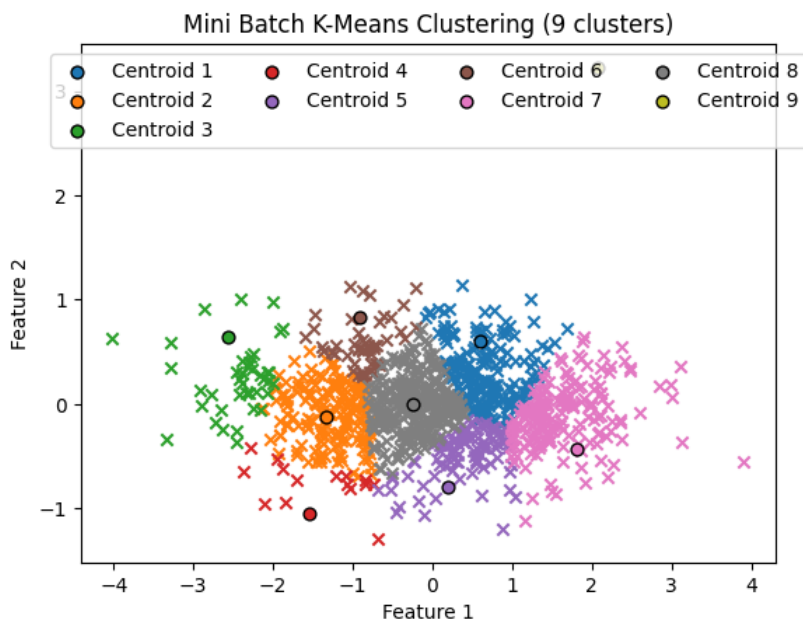


Figure 4.11

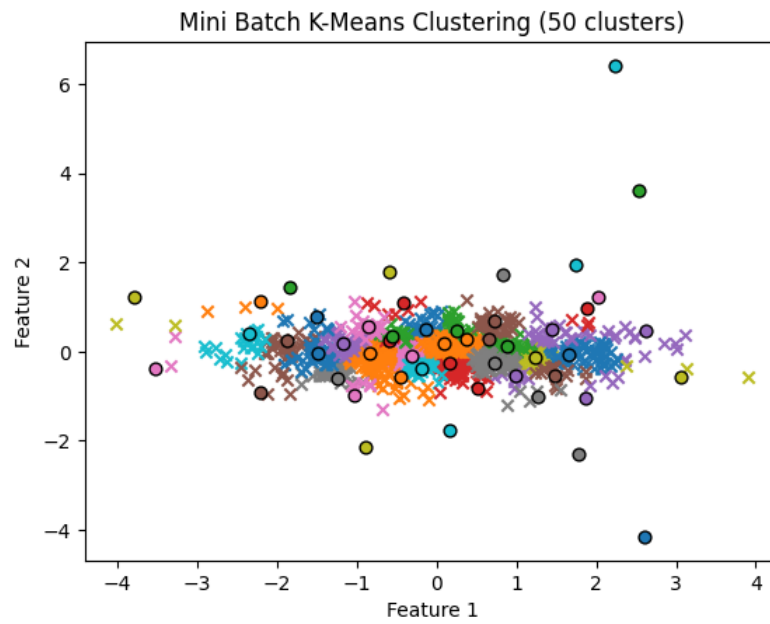


Figure 4.12

The best number of clusters, is elbow point in the plot of inertia with respect to number of clusters:

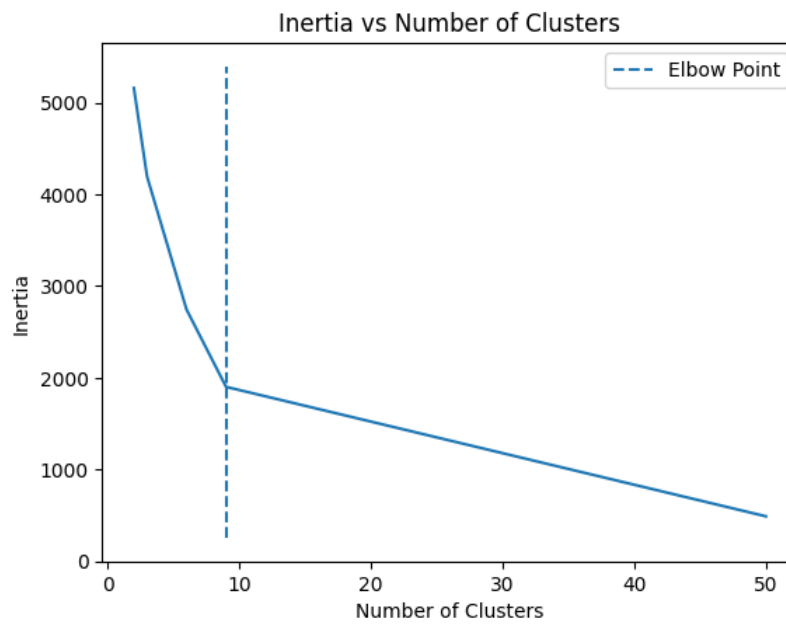


Figure 4.13

4.2 Gaussian Mixture Models

Gaussian Mixture Models (GMMs) are powerful probabilistic models used for clustering and density estimation. By combining multiple Gaussian components, GMMs can represent various data patterns and capture the underlying structure of the data. They have widespread applications in clustering, anomaly detection, image and speech recognition, and data generation. With the ability to estimate parameters using the Expectation-Maximization (EM) algorithm, GMMs provide valuable insights and analysis capabilities in diverse fields.

The Expectation-Maximization (EM) algorithm is commonly employed to estimate the parameters of Gaussian Mixture Models (GMMs), including the mean, covariance, and cluster weights. The EM algorithm iteratively refines these parameters by alternating between the expectation (E) step, where data points are probabilistically assigned to clusters, and the maximization (M) step, where the model parameters are updated based on the assignments.

Advantages:

- **Flexibility:** GMMs can identify clusters of various shapes and sizes, making them suitable for datasets with complex distributions and overlapping clusters.
- **Probabilistic Assignments:** Unlike K-means, which yields hard cluster assignments, GMMs provide soft assignments, which are more informative and useful for uncertainty estimation.
- **Outlier Robustness:** GMMs are less sensitive to outliers since data points are assigned probabilities to clusters rather than being hard-assigned.
- **Model Selection:** GMMs offer a natural framework for model selection using techniques like the Bayesian Information Criterion (BIC) or cross-validation

Limitations and Disadvantages:

- **Computationally More Demanding:** Compared to K-means, GMMs are computationally more demanding due to the iterative EM algorithm, making them less efficient for large datasets.
- **Initialization Sensitivity:** GMMs are sensitive to initialization, and suboptimal initializations may lead to convergence to local optima.

Gaussian Mixture Models vs. K-means:

K-means, as mentioned in the introduction, is widely used due to its efficient time performance. However, it suffers from scalability issues when dealing with large datasets. Gaussian Mixture Models tackle the limitations of K-means. Instead of assigning a data point exclusively to a single cluster, GMMs assign probabilities to data points belonging to each cluster. This makes GMMs more suitable for datasets with complex

structures, overlapping clusters, or uncertain data points.

For finding the optimal number of components in a cluster, the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) are commonly used measures.

AIC is a statistical measure that quantifies the quality of a model relative to the number of parameters it uses. It balances the goodness of fit of the model with the complexity of the model. When comparing different models, the model with the lowest AIC is preferred.

BIC is another criterion used for model selection. Like the AIC, it also penalizes models for their complexity. However, the BIC applies a stronger penalty for model complexity compared to the AIC. When comparing different models, the model with the lowest BIC is preferred.

Both the AIC and BIC are useful tools in model selection and determining the optimal number of components in a cluster.

4.2.1 Evaluation with Deep Features

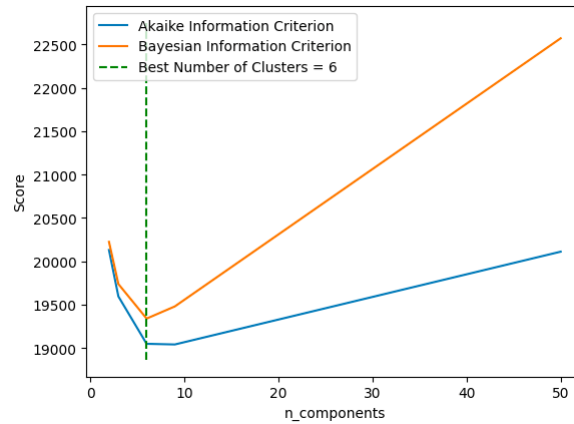


Figure 4.14: AIC and BIC

4.2.2 Evaluation with Handcraft Features

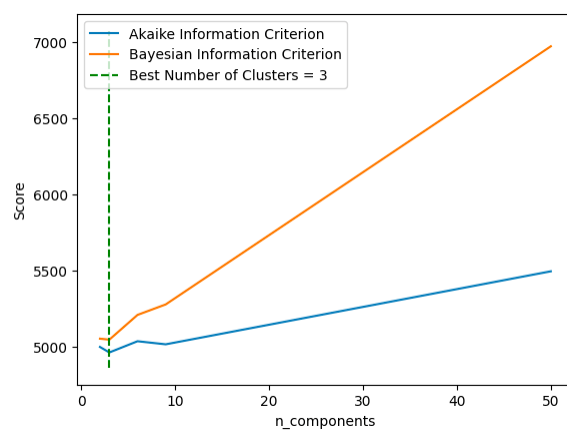


Figure 4.15: AIC and BIC

References

- [1] Zey ang. De-fake: Detection and attribution of fake images generated by text-to-image generation models. <https://arxiv.org/abs/2210.06998>, 13 Oct 2022.
- [2] AmiyaRanjanRout. Advantages and disadvantages of logistic regression. <https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/>, 10 Jan, 2023.
- [3] Hucker Marius. Multiclass classification with support vector machines (svm), dual problem and kernel functions. <https://towardsdatascience.com/multiclass-classification-with-support-vector-machines-svm-kernel-trick-kernel-functions-f9d5377d6f02>, Jun 9, 2020.
- [4] Great Learning Team. Random forest algorithm in machine learning: An overview. <https://www.mygreatlearning.com/blog/random-forest-algorithm/>, Jun 13, 2023.
- [5] GeeksforGeeks. K means clustering–introduction. <https://www.geeksforgeeks.org/k-means-clustering-introduction/>, 24 Mar, 2023.
- [6] unknown. Pros and cons of k-means clustering. <https://prosancons.com/education/pros-and-cons-of-k-means-clustering/>, November 24, 2018.