

Project 3: Exploring Different Classifiers & Regression Methods

Group 4:

Daniel González Muela

Francisco Boudagh

Purusothaman Seenivasen

Sky Sunsaksawat

Messaoud Shaker

MVE441 Statistical Learning for Big Data

23th May 2024.



CHALMERS
UNIVERSITY OF TECHNOLOGY

5 different classifiers

1. CNN

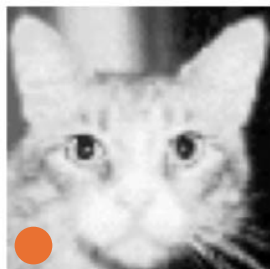
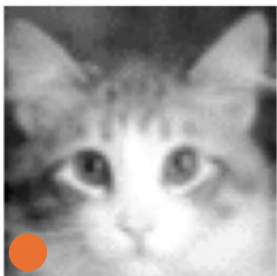
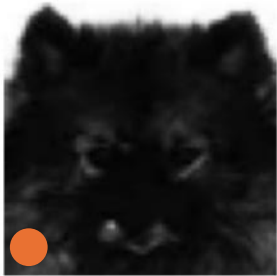
Accuracies

Cats 79%

★ Dogs 100%

15	0
4	21

Missclassified:



2. SVM

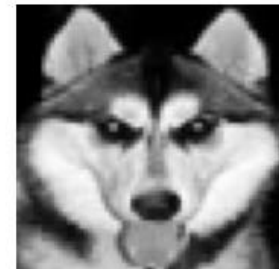
Accuracies

★ Cats 94%

Dogs 86%

17	3
1	19

Missclassified:



3. K-NN

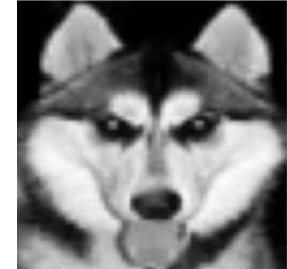
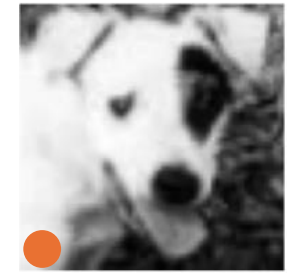
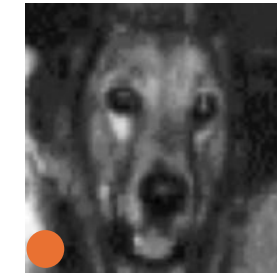
Accuracies

Cats 75%

Dogs 79%

12	5
4	19

Missclassified:



5 Different Classifiers

4. Logistic Regression

Accuracies

Cats 75%

Dogs 80%

15	4
5	16

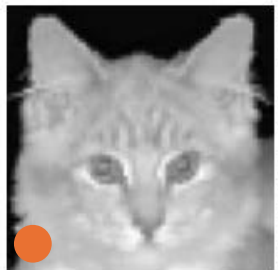
Missclassified:



Pred: Cat, True: Dog



Pred: Dog, True: Cat



Pred: Cat, True: Dog



Pred: Dog, True: Cat

5. Naive Bayes

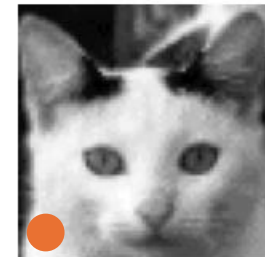
Accuracies

Cats 74%

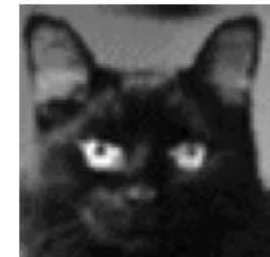
Dogs 76%

14	5
5	16

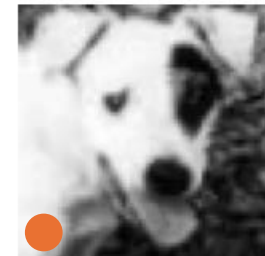
Missclassified:



Pred: Cat, True: Dog



Pred: Dog, True: Cat



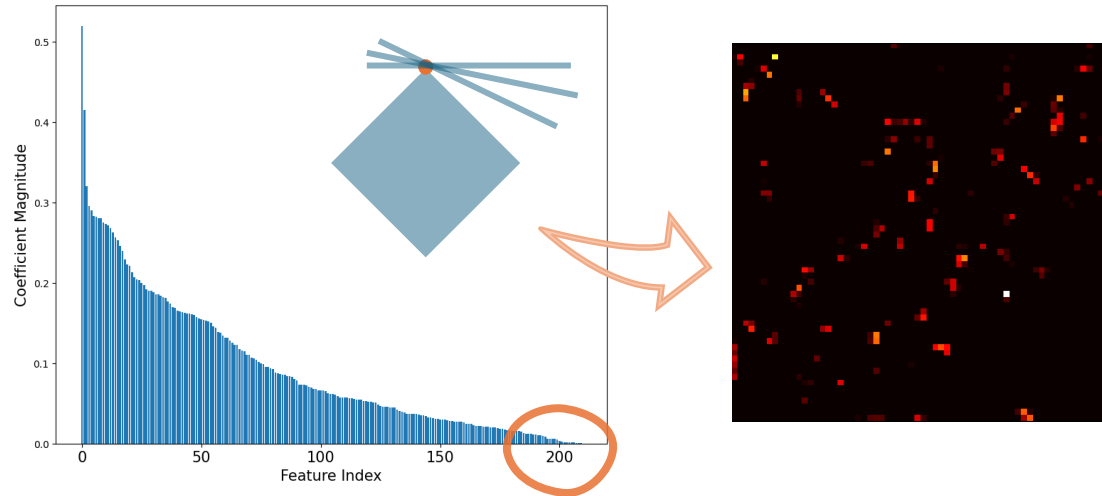
Pred: Dog, True: Cat



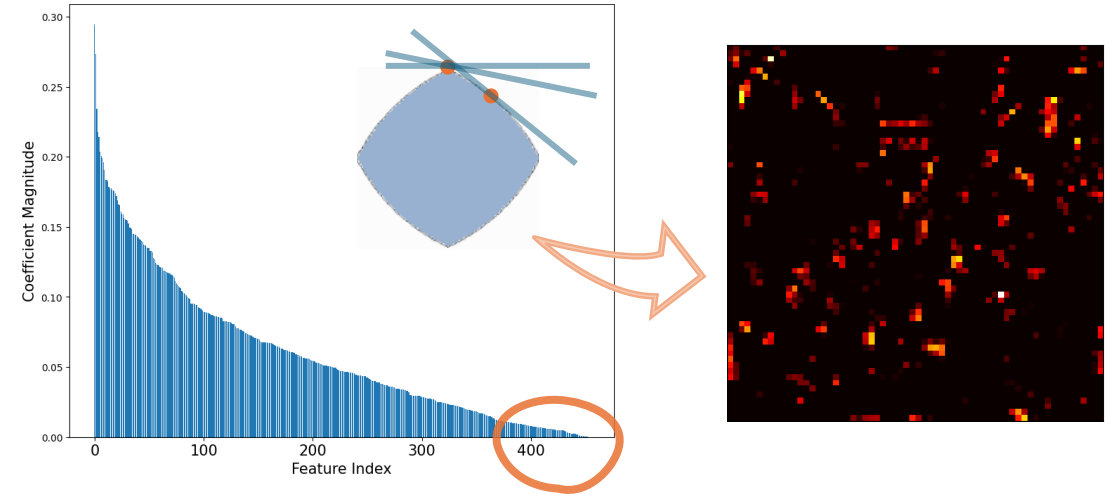
Pred: Cat, True: Dog

Feature Importance

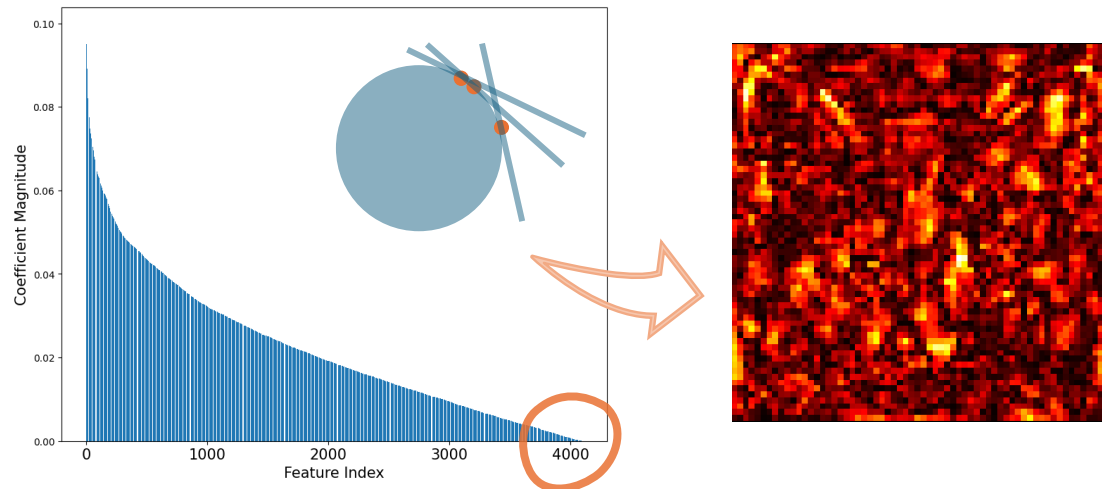
Lasso Logistic Regression



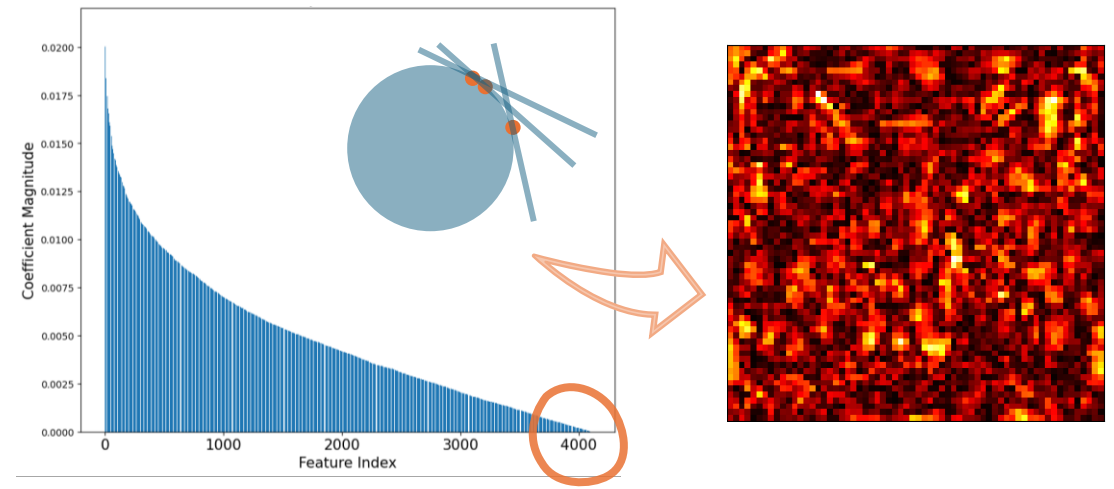
Elastic Net Logistic Regression



Ridge Logistic Regression



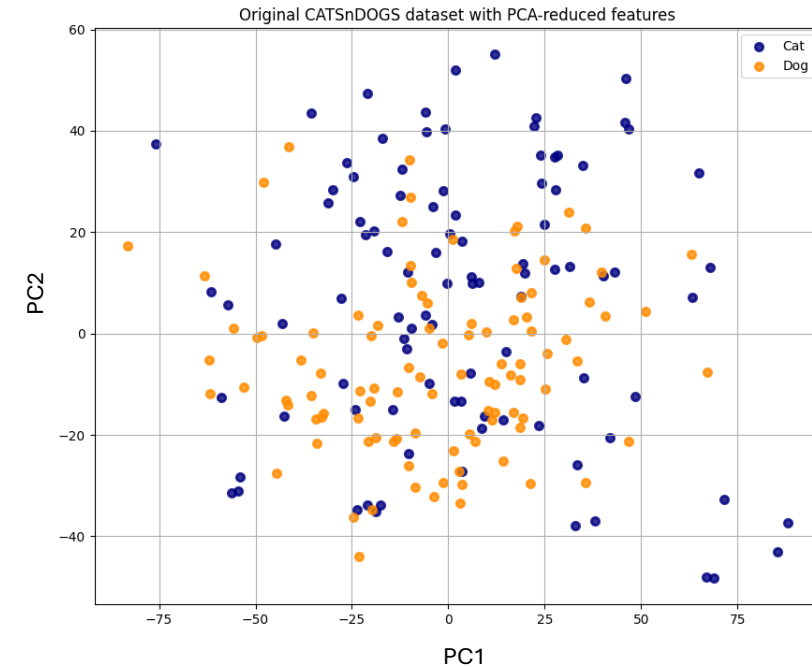
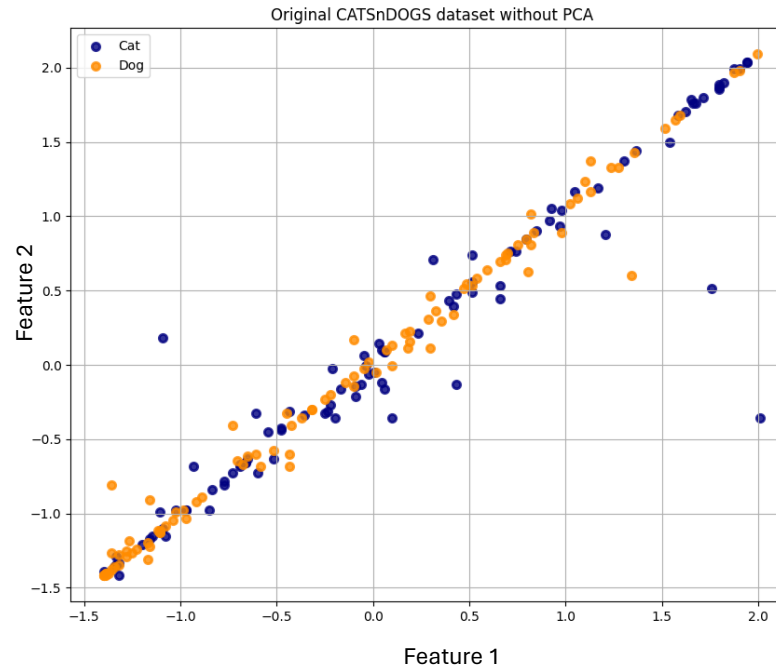
Ridge SVM



Clustering

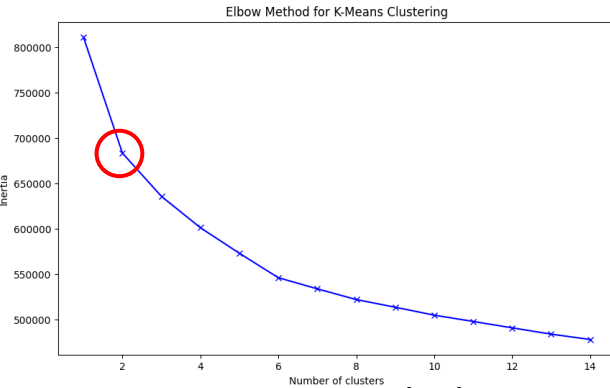
Method setup

- Processing data :
 - Normalize data (StandardScaler)
 - Feature reduction (PCA)
- Clustering method :
 - K-means
 - Partition around medoids (PAM) or K-medoids
- Analyze method :
 - Elbow heuristics
 - Silhouette score
 - Normal mutual information(NMI)

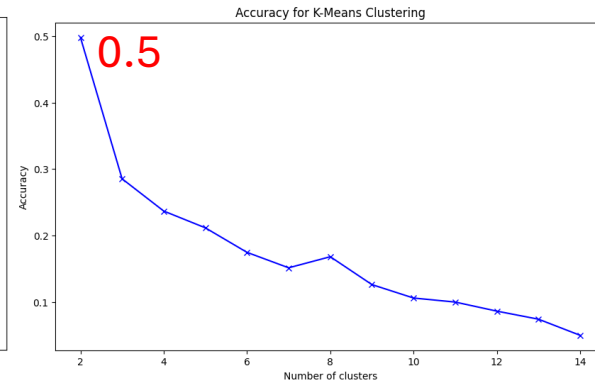


Clustering

K-means Clustering

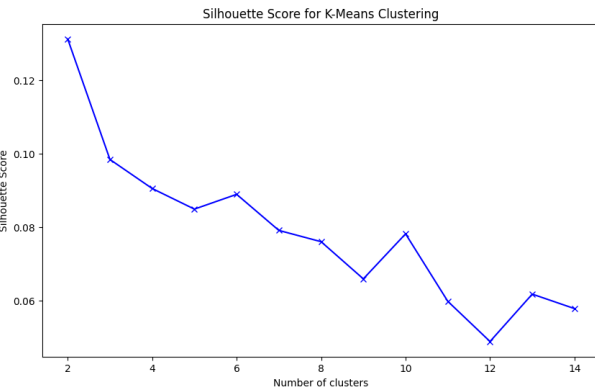
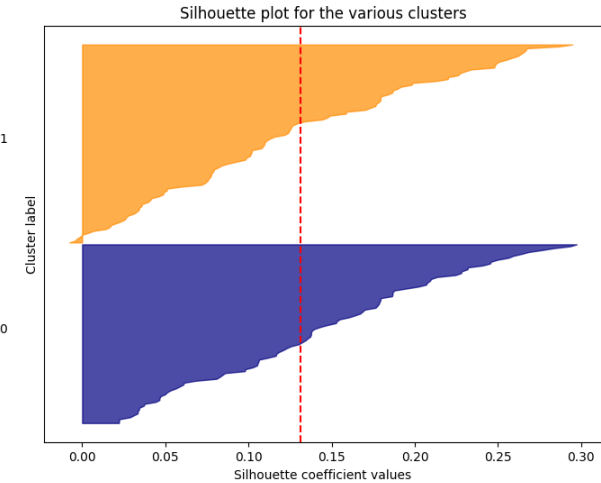
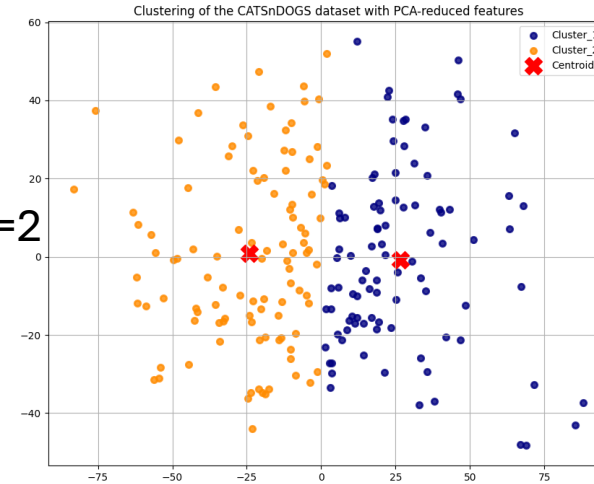


Elbow heuristic

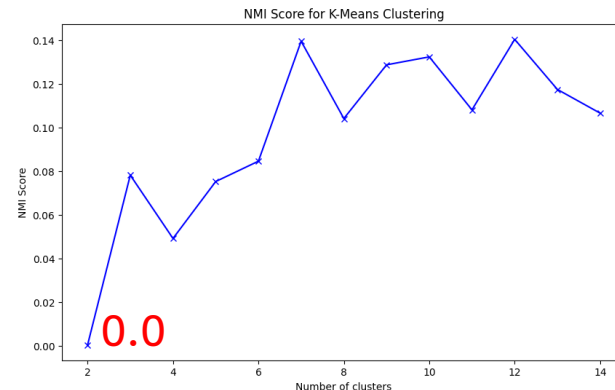


Accuracy

K=2

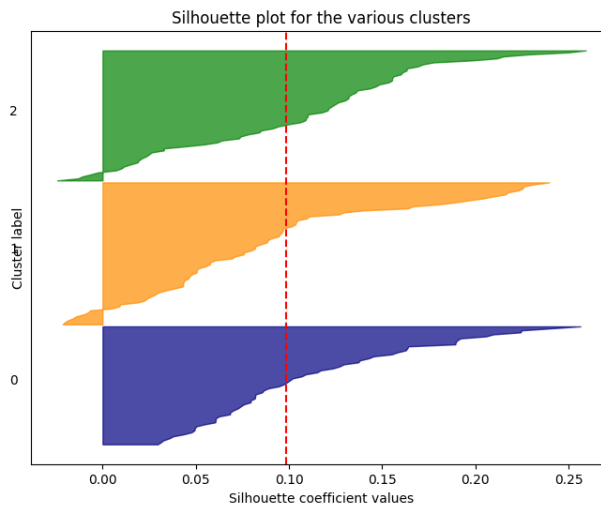
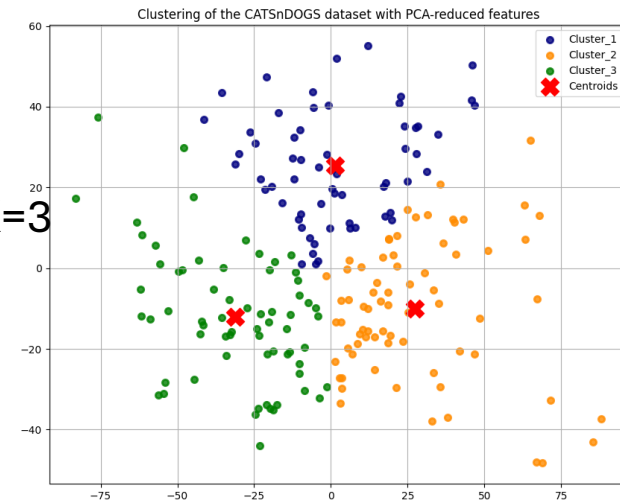


Silhouette score (mean)



NMI

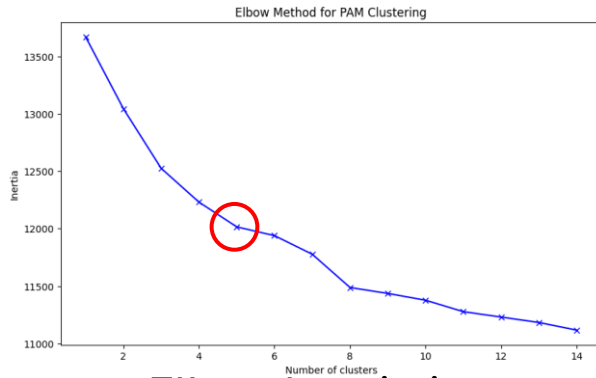
K=3



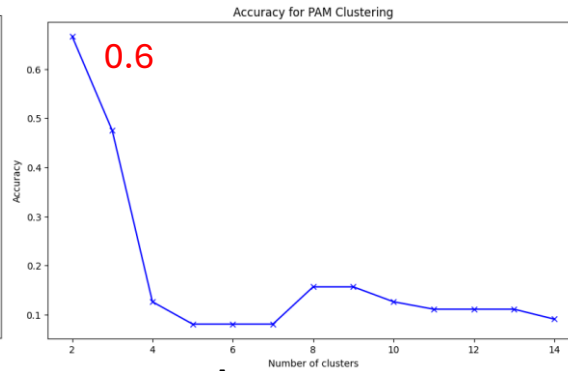
The clusters seem to be random according to the class labels(not agree with labels),
Increasing K has minor impact on overlapping

Clustering

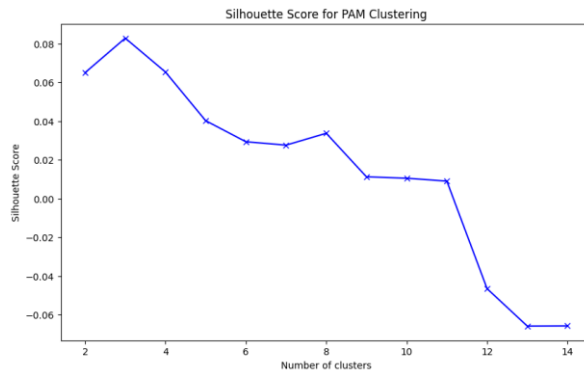
Partition around medoids (PAM) or K-medoids



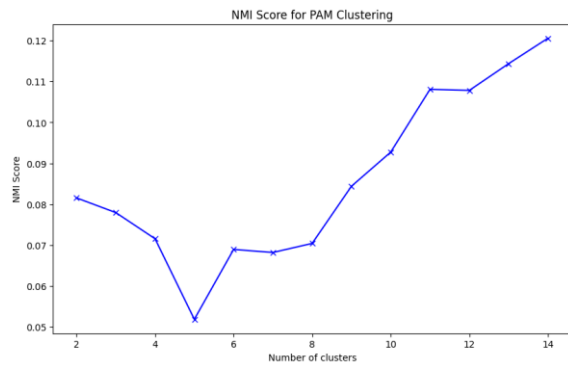
Elbow heuristic



Accuracy

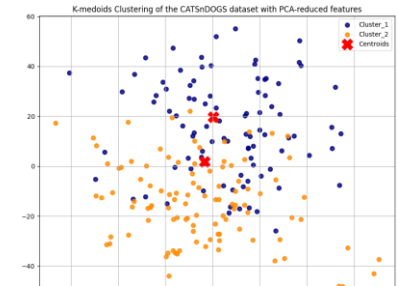
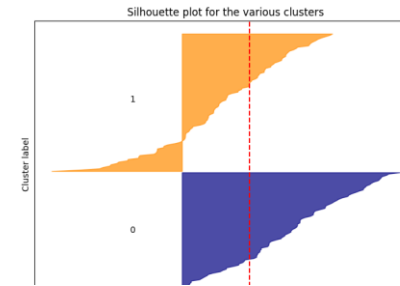


Silhouette score (mean)

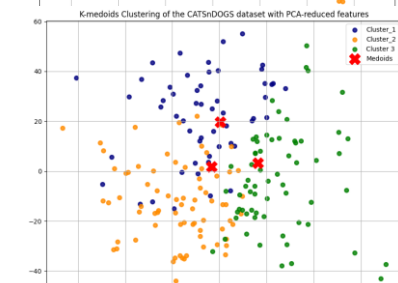


NMI

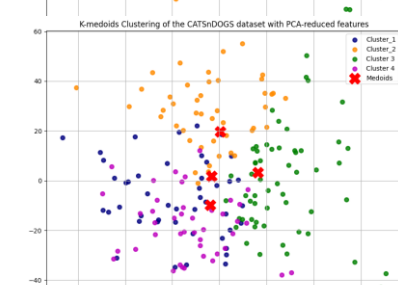
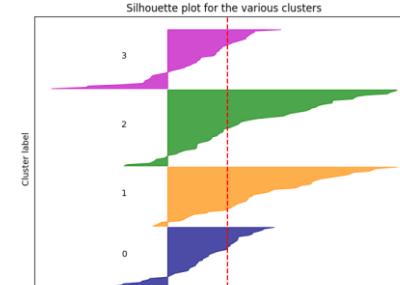
K=2



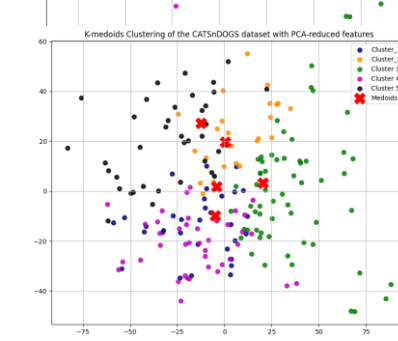
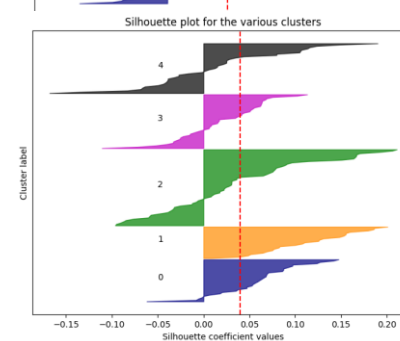
K=3



K=4



K=5

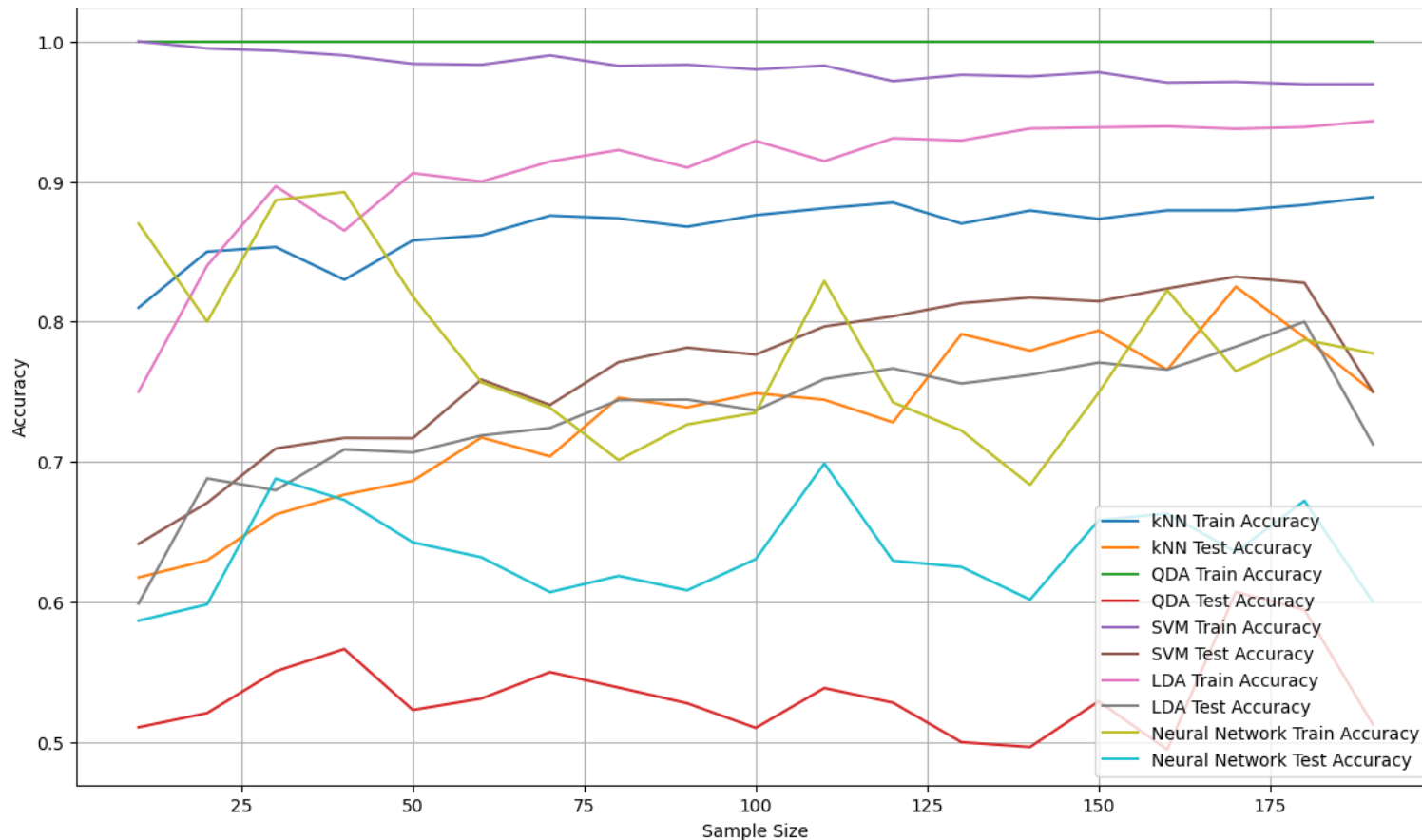


Accuracy gradually higher than K-mean due to cluster overlapping (similar to dataset)

Increasing K has more impact on overlapping than K-means (more negative silhouette value)

Simulation of Cats and Dogs Dataset for various sample sizes

Accuracy vs. Sample Size



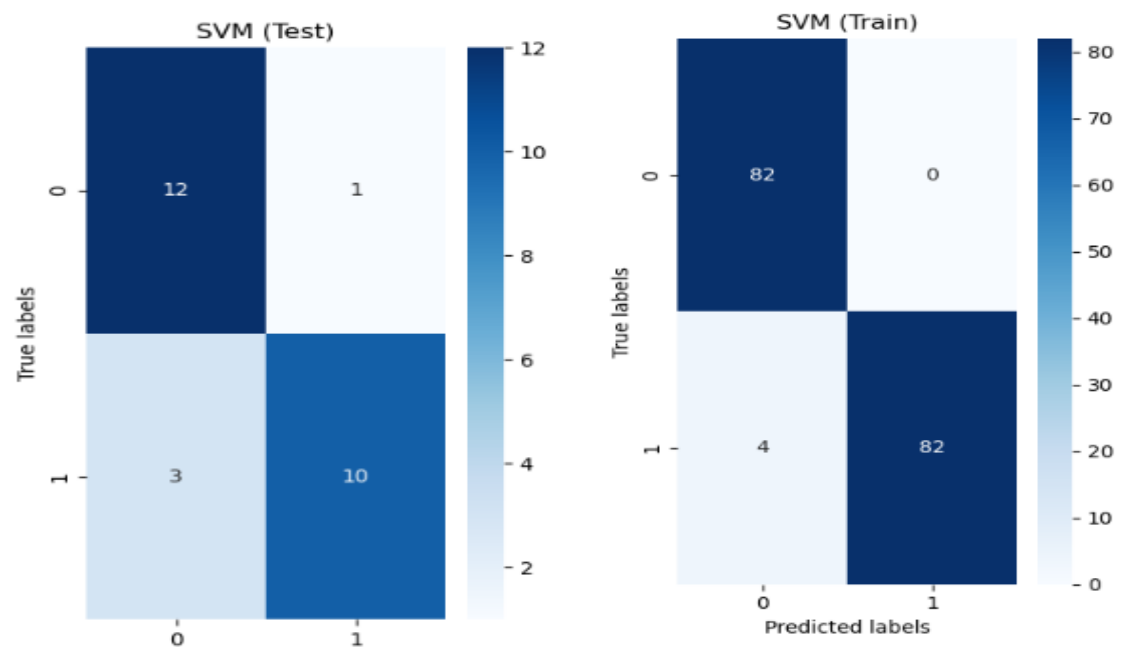
The performance of different models on the Cats and Dogs dataset

Model	Avg Train Acc.	Avg Test Acc.	Fit Quality
SVM	High	High	Good Fit
LDA	High	High	Good Fit
KNN	High	slightly lower	Slight Overfit
MLP	Moderate	Above moderate	Good fit
QDA	High	lower	Overfit

SVM, small differences between train and test accuracies,.
LDA, Small differences between train and test accuracies.
KNN, Moderate differences between train and test accuracies.
MLP, the difference between training and testing accuracies is not large.
QDA, Large differences between train and test accuracies.

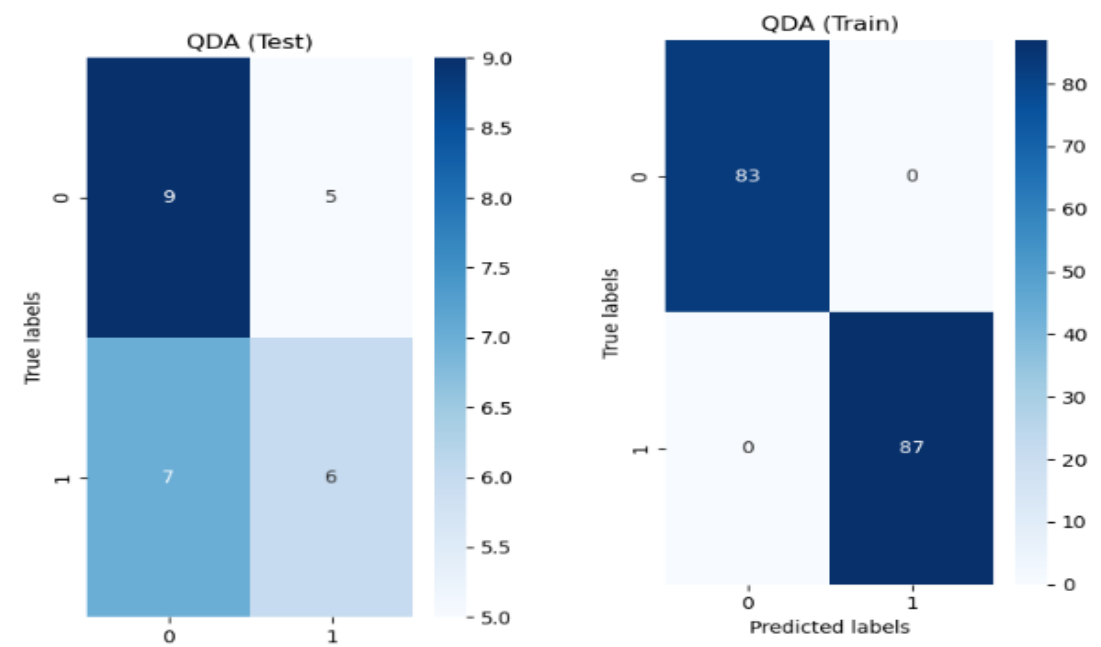
Confusion Matrices for Sample Size 170 (Best & Worst)

SVM



SVM with RBF kernel is performing well both in training and testing. This may be due to non-linear decision boundaries

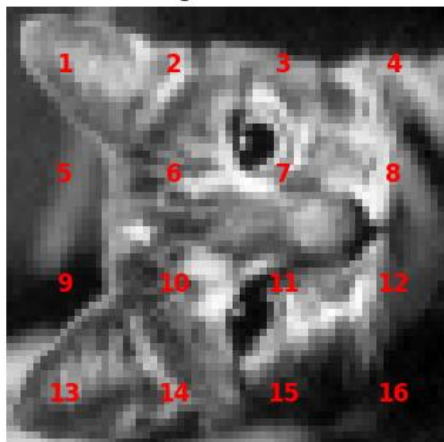
QDA



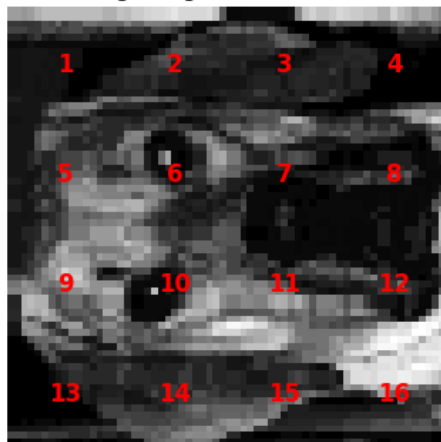
QDA assumes that each class follows a Gaussian distribution with its own covariance matrix. If this assumption is not met, the model may not generalize well to unseen data.

Simulation of cat and dog dataset by patching and evaluating performance

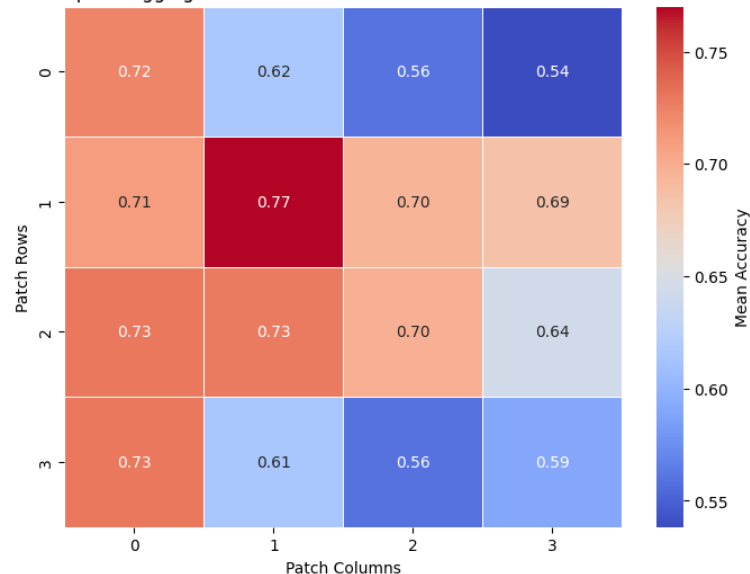
Cat Image with Patches



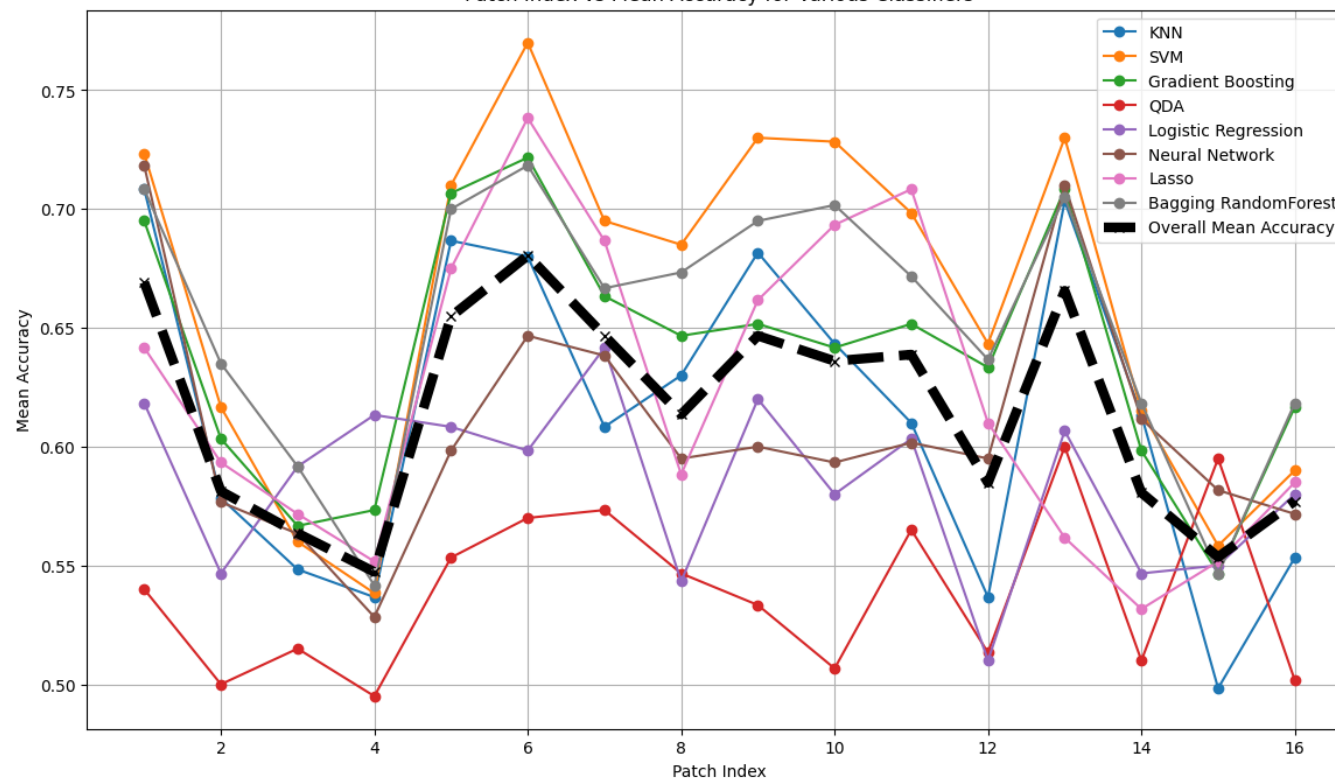
Dog Image with Patches



Heatmap of Bagging RandomForest Mean Accuracies Across 4x4 Patches

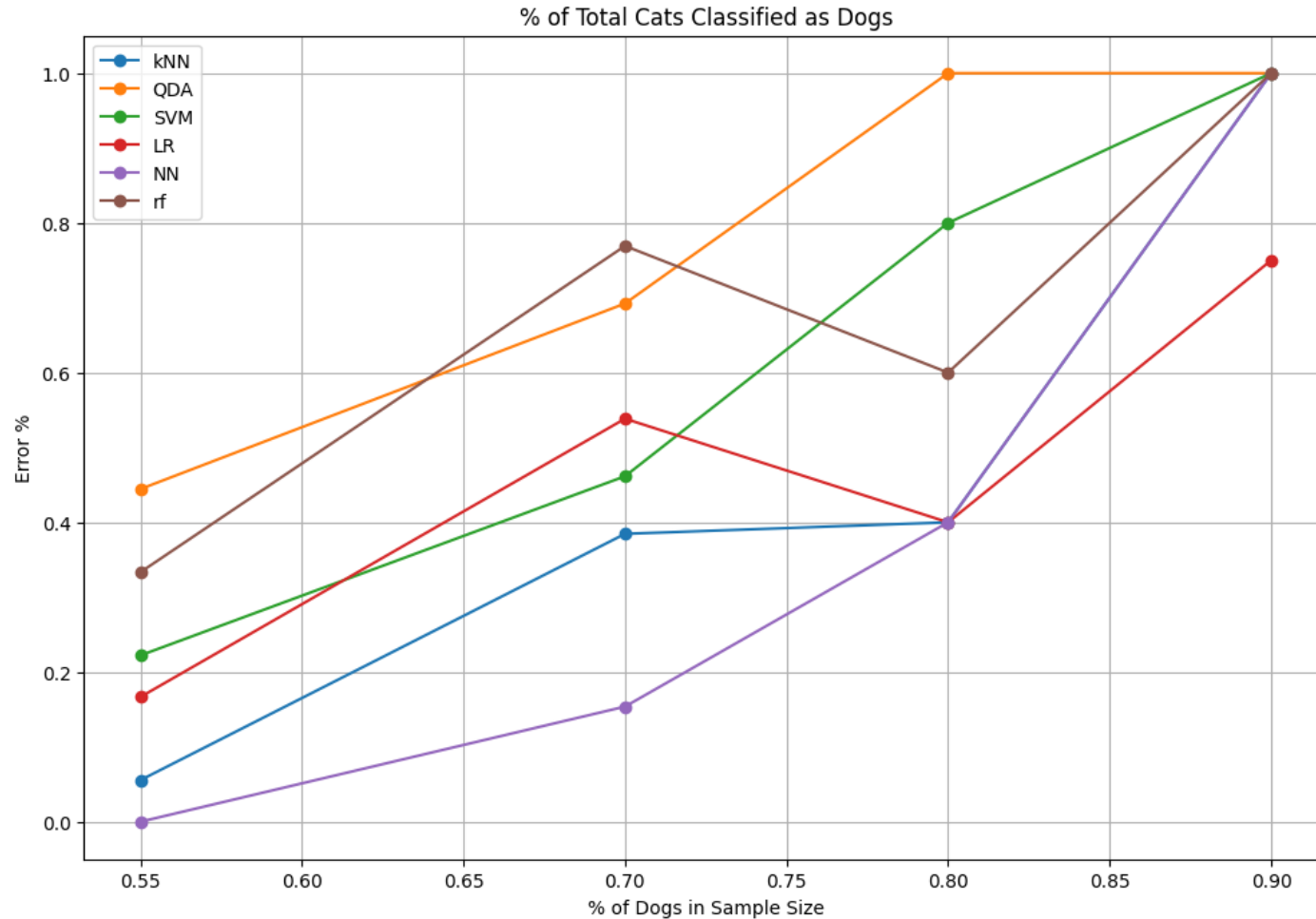


Patch Index vs Mean Accuracy for Various Classifiers



- Best method: SVM (RBF kernel)
- Best patch:
 - Patch 6 & 10 (dog's eyes);
 - Patch 1 & 13 (ears);
 - patch 7 & 11 (cat's eyes)

Portion of dogs in sample



Finally, we increased the % of dogs in the dataset

Is there any algorithm that would manage oversampling better?

Apparently not. We observe a constant increase across all methods