Using PCA as a dimensionality reduction technique

**Neermita Bhattacharya B22CS092**

# Using PCA + KNN:

## **n-components = 250**

**…**

Best estimator found by grid search:

KNeighborsClassifier(n\_neighbors=2, weights='distance')

Accuracy = 0.25

Classification Report:

precision recall f1-score support

0 0.24 0.42 0.31 1000

1 0.90 0.09 0.16 1000

2 0.20 0.30 0.24 1000

3 0.23 0.13 0.17 1000

4 0.19 0.51 0.28 1000

5 0.33 0.09 0.15 1000

6 0.31 0.19 0.24 1000

7 0.73 0.10 0.18 1000

8 0.25 0.62 0.36 1000

9 0.77 0.04 0.07 1000

## **n-components = 100:**

Best estimator found by grid search:

KNeighborsClassifier(n\_neighbors=3, weights='distance')

Accuracy: 0.3409

Classification Report:

precision recall f1-score support

0 0.39 0.52 0.45 1000

1 0.81 0.19 0.31 1000

2 0.22 0.39 0.28 1000

3 0.29 0.18 0.22 1000

4 0.25 0.54 0.34 1000

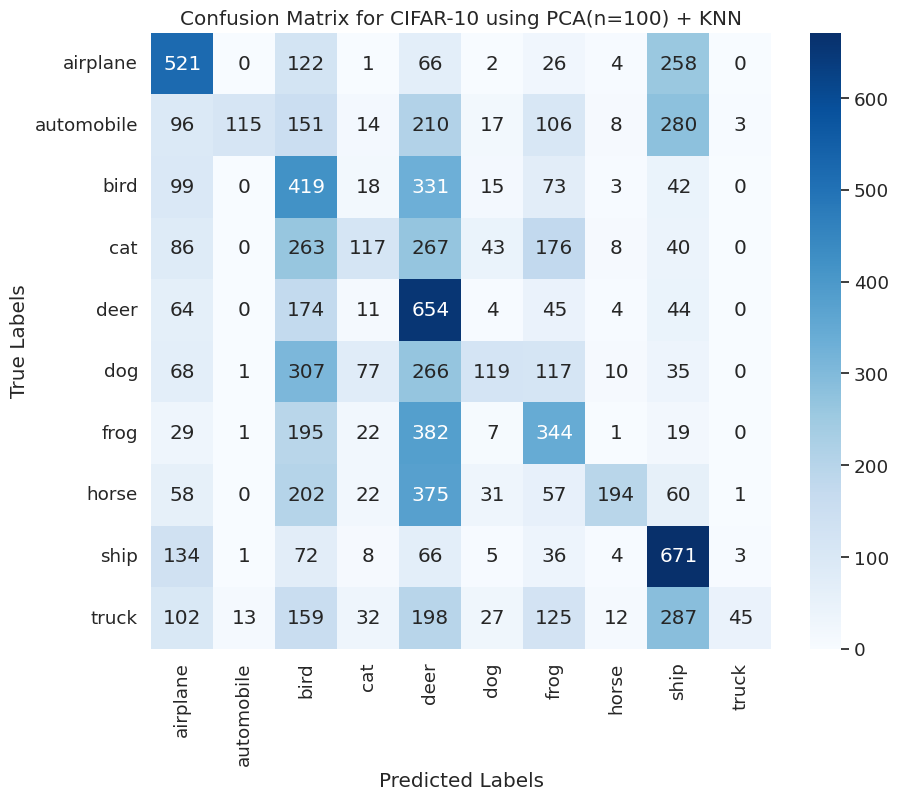
5 0.38 0.18 0.25 1000

6 0.32 0.39 0.35 1000

7 0.66 0.28 0.39 1000

8 0.39 0.61 0.48 1000

9 0.69 0.12 0.21 1000



## **n-components = 80:**

**Best estimator found by grid search:**

**KNeighborsClassifier(n\_neighbors=4, weights='distance')**

Accuracy: 0.366

Classification Report:

precision recall f1-score support

0 0.43 0.55 0.48 1000

1 0.80 0.23 0.36 1000

2 0.24 0.45 0.31 1000

3 0.35 0.19 0.25 1000

4 0.26 0.54 0.35 1000

5 0.40 0.20 0.26 1000

6 0.33 0.45 0.38 1000

7 0.66 0.30 0.41 1000

8 0.43 0.61 0.51 1000

9 0.72 0.14 0.23 1000

## **n-components = 60:**

Best estimator found by grid search:

KNeighborsClassifier(n\_neighbors=7, weights='distance')

Accuracy: 0.3908

Classification Report:

precision recall f1-score support

0 0.44 0.56 0.49 1000

1 0.70 0.28 0.41 1000

2 0.26 0.44 0.33 1000

3 0.33 0.18 0.23 1000

4 0.29 0.54 0.38 1000

5 0.44 0.22 0.30 1000

6 0.33 0.51 0.40 1000

7 0.63 0.36 0.46 1000

8 0.45 0.62 0.52 1000

9 0.69 0.19 0.30 1000

## **n-components = 50:**

**Best estimator found by grid search:**

**KNeighborsClassifier(n\_neighbors=10, weights='distance')**

**Accuracy: 0.401**

Classification Report:

**precision recall f1-score support**

**0 0.47 0.57 0.51 1000**

**1 0.75 0.28 0.41 1000**

**2 0.27 0.44 0.33 1000**

**3 0.36 0.16 0.22 1000**

**4 0.29 0.59 0.39 1000**

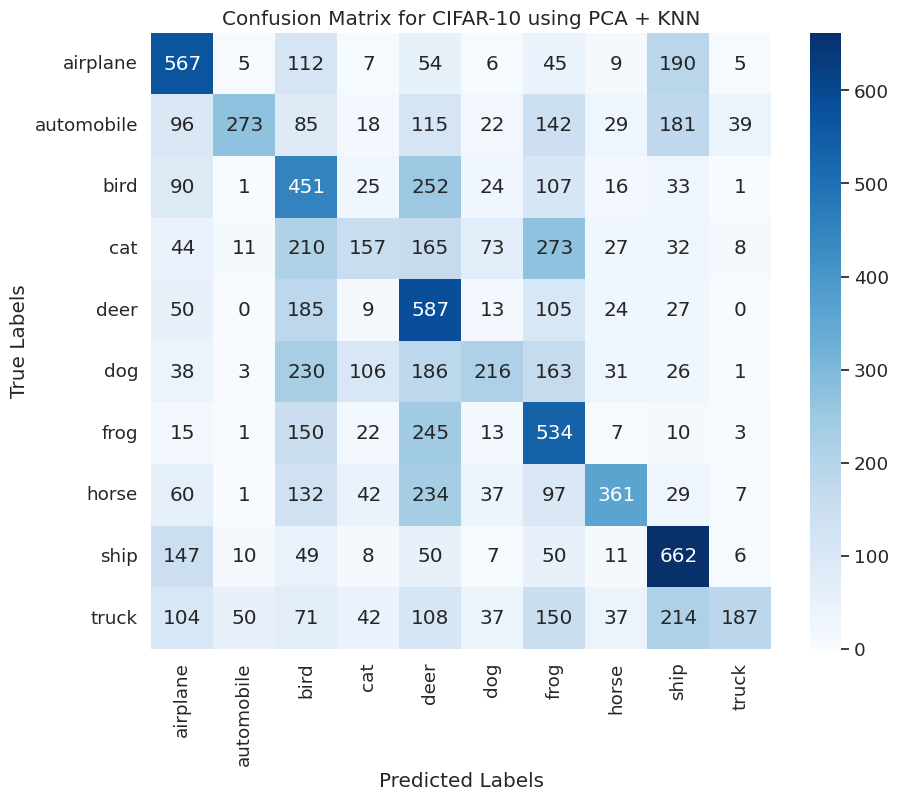
**5 0.50 0.21 0.30 1000**

**6 0.32 0.53 0.40 1000**

**7 0.68 0.37 0.48 1000**

**8 0.47 0.65 0.55 1000**

**9 0.73 0.20 0.32 1000**



# Using PCA + Decision Trees:

## **n-components=100**

Best Parameters: {'max\_depth': 10}

accuracy = 0.31

Classification Report:

precision recall f1-score support

0 0.45 0.43 0.44 1000

1 0.35 0.34 0.34 1000

2 0.23 0.25 0.24 1000

3 0.21 0.14 0.17 1000

4 0.31 0.24 0.27 1000

5 0.29 0.27 0.28 1000

6 0.24 0.45 0.31 1000

7 0.33 0.21 0.26 1000

8 0.43 0.45 0.44 1000

9 0.35 0.35 0.35 1000

## **n-components=50**

Best Parameters: {'max\_depth': 10}

accuracy = 0.32

Classification Report:

precision recall f1-score support

0 0.41 0.45 0.43 1000

1 0.36 0.34 0.35 1000

2 0.30 0.17 0.21 1000

3 0.22 0.18 0.20 1000

4 0.23 0.28 0.25 1000

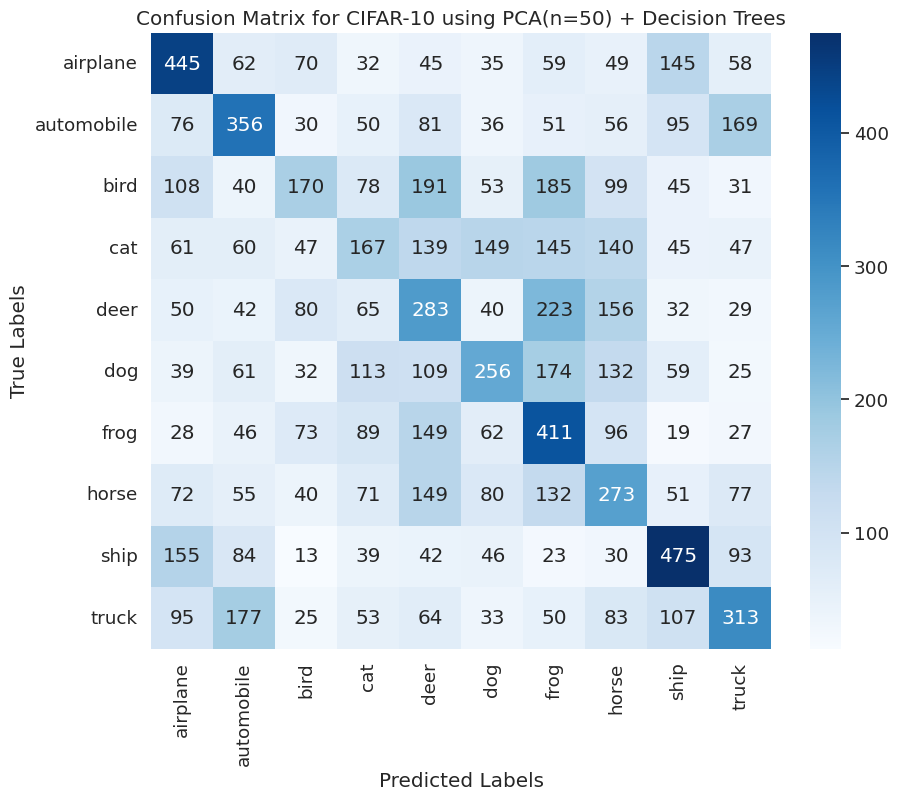
5 0.32 0.25 0.28 1000

6 0.28 0.42 0.34 1000

7 0.25 0.28 0.26 1000

8 0.44 0.48 0.46 1000

9 0.36 0.33 0.34 1000



## **n-components=250**

**Accuracy = 0.2579**

# Using PCA +AdaBoost:

## **n-components=100**

accuracy = 0.37

Classification Report:

precision recall f1-score support

0 0.41 0.43 0.42 1000

1 0.44 0.40 0.42 1000

2 0.26 0.19 0.22 1000

3 0.28 0.17 0.21 1000

4 0.29 0.27 0.28 1000

5 0.34 0.30 0.32 1000

6 0.34 0.57 0.43 1000

7 0.40 0.35 0.37 1000

8 0.41 0.55 0.47 1000

9 0.43 0.45 0.44 1000

## **n-components=50**

accuracy = 0.35

Classification Report:

precision recall f1-score support

0 0.40 0.44 0.42 1000

1 0.40 0.35 0.38 1000

2 0.25 0.17 0.20 1000

3 0.29 0.16 0.21 1000

4 0.29 0.26 0.27 1000

5 0.35 0.33 0.34 1000

6 0.33 0.54 0.41 1000

7 0.35 0.31 0.33 1000

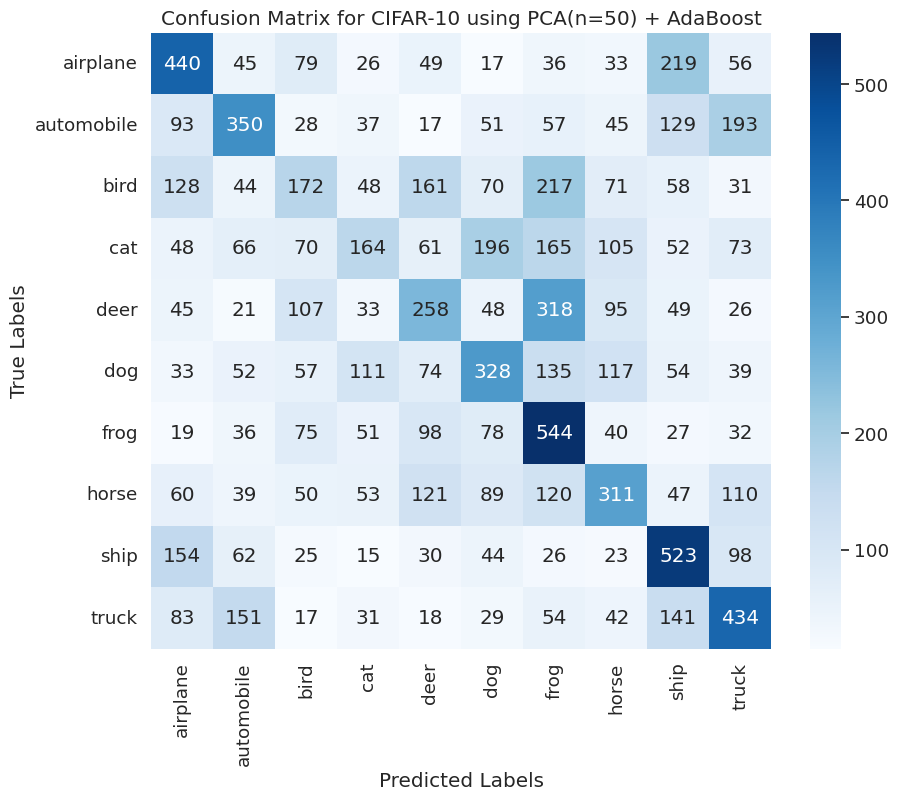
8 0.40 0.52 0.45 1000

9 0.40 0.43 0.41 1000

accuracy 0.35 10000

macro avg 0.35 0.35 0.34 10000

weighted avg 0.35 0.35 0.34 10000



# Using PCA + Random Forests:

## **n-components=100**

accuracy = 0.45

Classification Report:

precision recall f1-score support

0 0.54 0.55 0.55 1000

1 0.52 0.59 0.55 1000

2 0.35 0.30 0.32 1000

3 0.29 0.24 0.26 1000

4 0.43 0.40 0.41 1000

5 0.37 0.36 0.37 1000

6 0.46 0.54 0.49 1000

7 0.51 0.44 0.47 1000

8 0.54 0.61 0.57 1000

9 0.45 0.50 0.47 1000

## **n-components=50**

accuracy = 0.46

Classification Report:

precision recall f1-score support

0 0.55 0.57 0.56 1000

1 0.52 0.57 0.54 1000

2 0.37 0.30 0.33 1000

3 0.32 0.29 0.31 1000

4 0.42 0.38 0.40 1000

5 0.39 0.38 0.39 1000

6 0.46 0.55 0.50 1000

7 0.50 0.46 0.48 1000

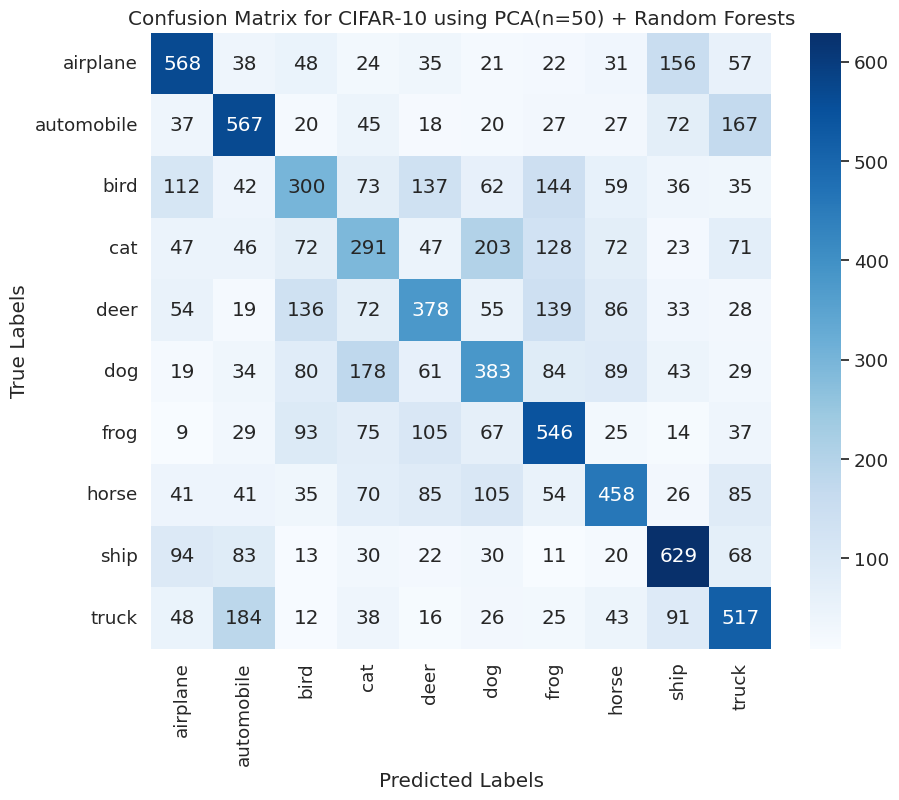
8 0.56 0.63 0.59 1000

9 0.47 0.52 0.49 1000

accuracy 0.46 10000

macro avg 0.46 0.46 0.46 10000

weighted avg 0.46 0.46 0.46 10000



| **PCA n--components** | **ACCURACY** | | | |
| --- | --- | --- | --- | --- |
|  | **KNN** | **Decision Trees** | **AdaBoost** | **Random Forests** |
| **250** | **25%** | **31%** | **-** | **-** |
| **100** | **34.1%** | **31%** | **37%** | **45%** |
| **50** | **40.1%** | **32%** | **35%** | **46%** |