CS 251, MACHINE LEARNING

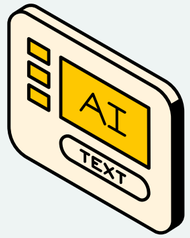
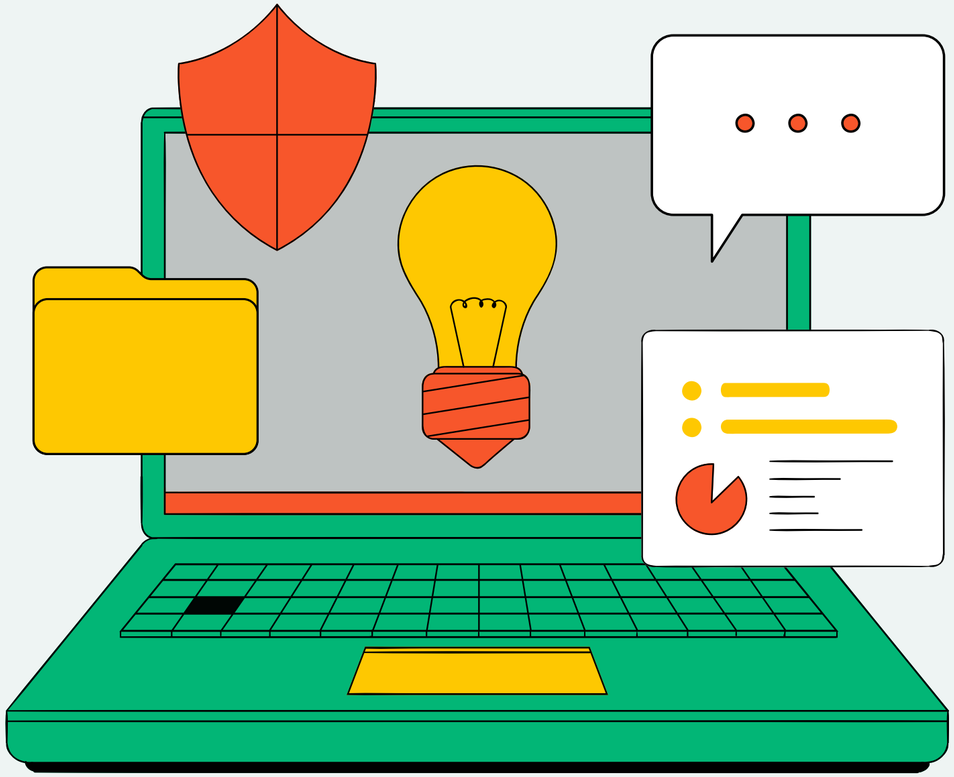
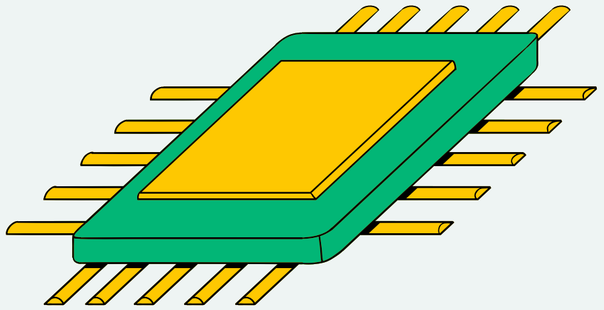


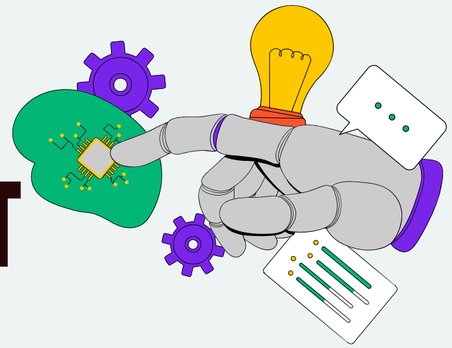
IMAGE CLASSIFICATION

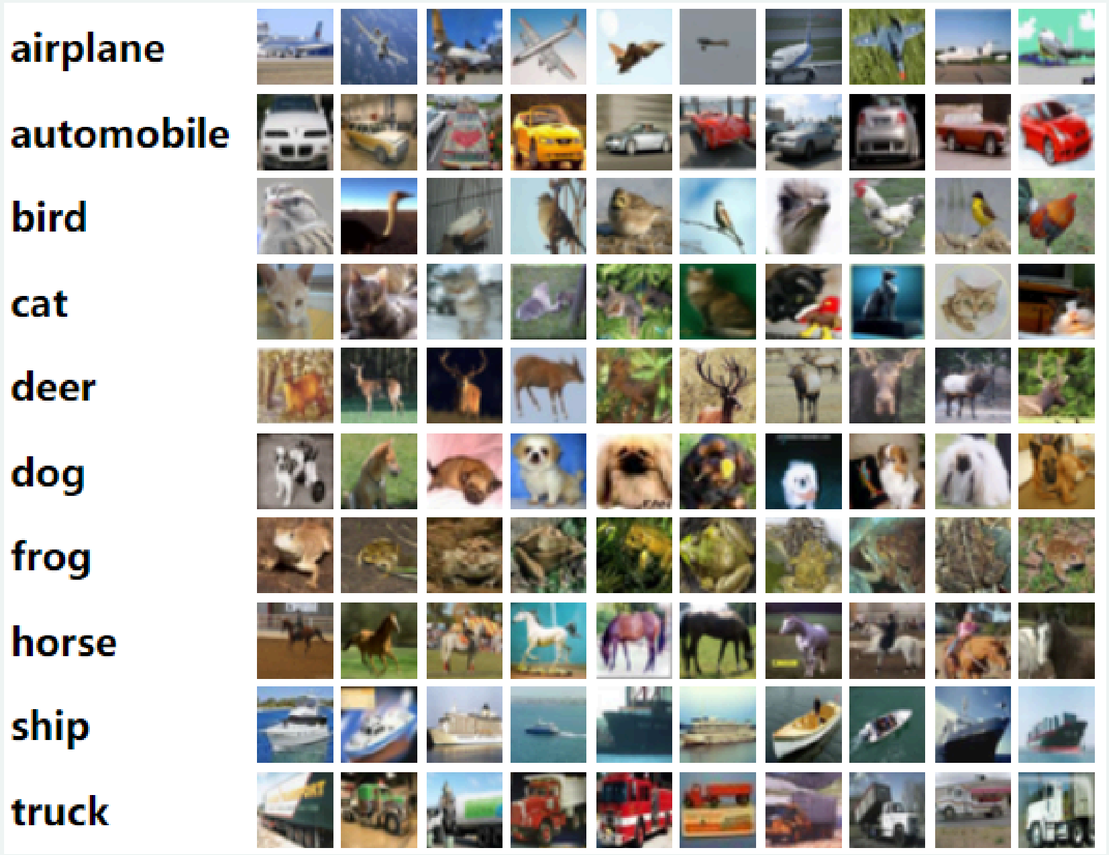
Group Members:
Narek Ghukasyan, Shushan Gevorgyan, Armen Madoyan, Yeva Stepanyan





DATASET



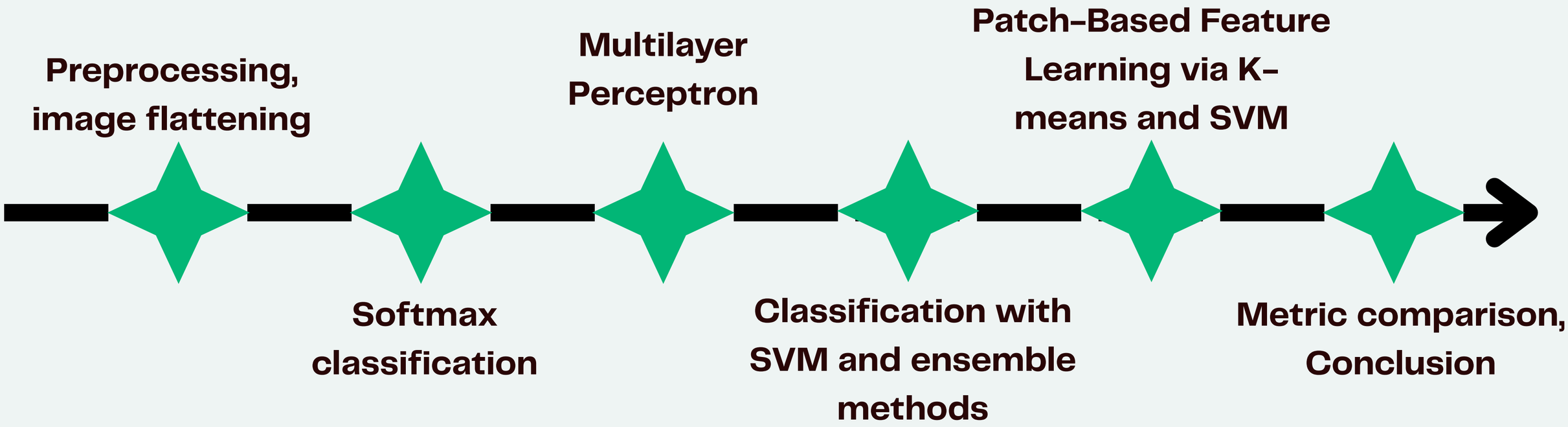


CIFAR-10

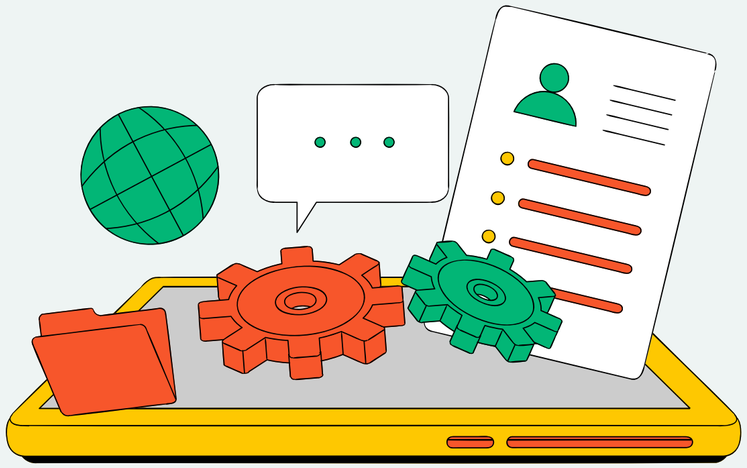
32x32 colour images in 10 classes

60.000 images, 6000 image per class
(the dataset is balanced)

APPLIED STRATEGIES



PREPROCESSING STEPS



PCA

Flattening

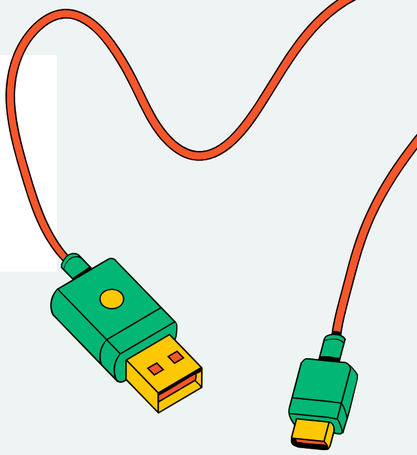
Whitening

Labeling

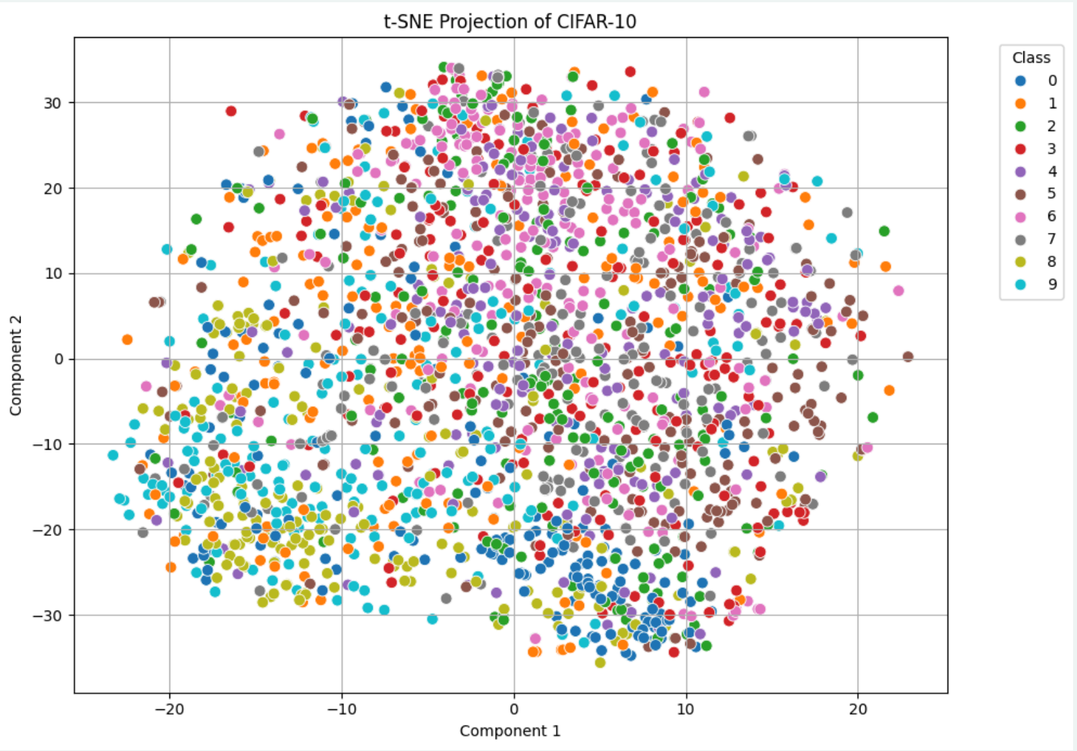
Normalizing

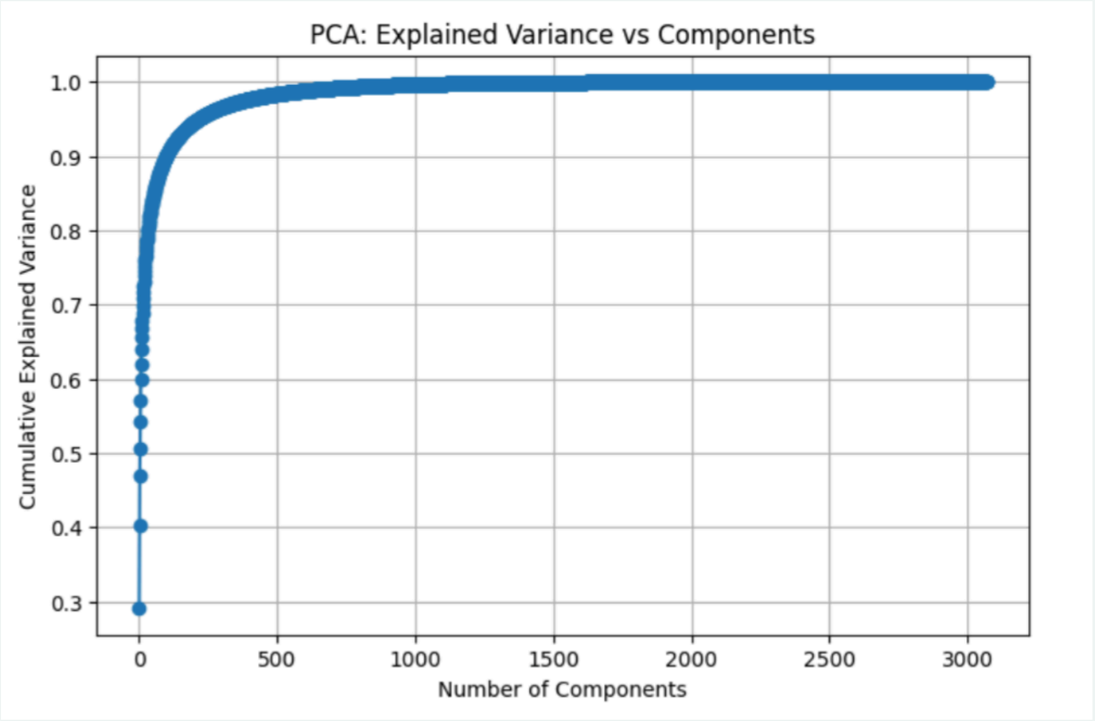
0: 'airplane', 1: 'automobile', 2: 'bird', 3: 'cat', 4: 'deer', 5: 'dog', 6: 'frog', 7: 'horse', 8: 'ship', 9: 'truck'

```
array([[[178, 176, 189],
        [178, 176, 189],
        [178, 176, 189],
```



```
data['train'][2]['img']
```

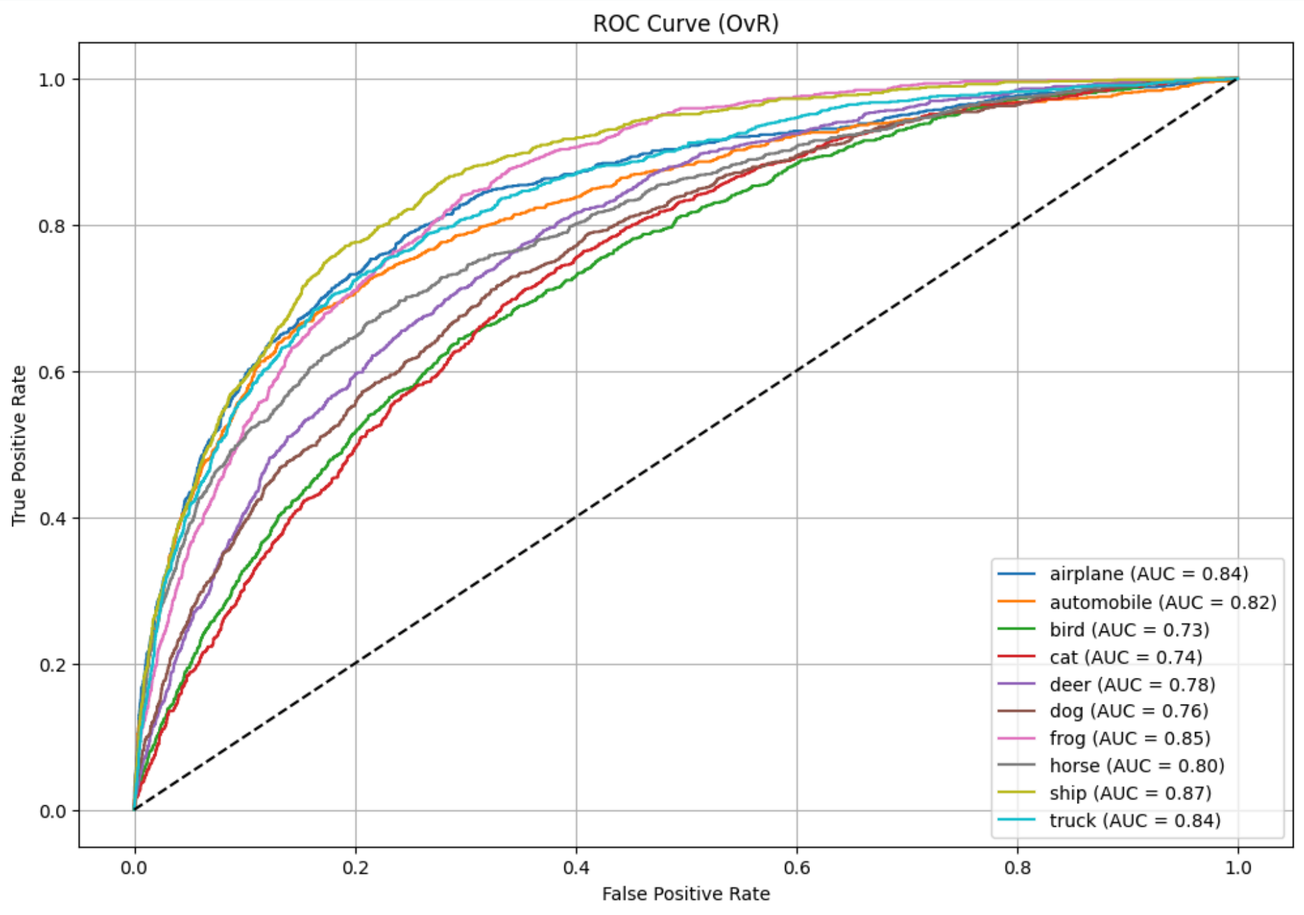




Optimal number of components for 95% variance: 217

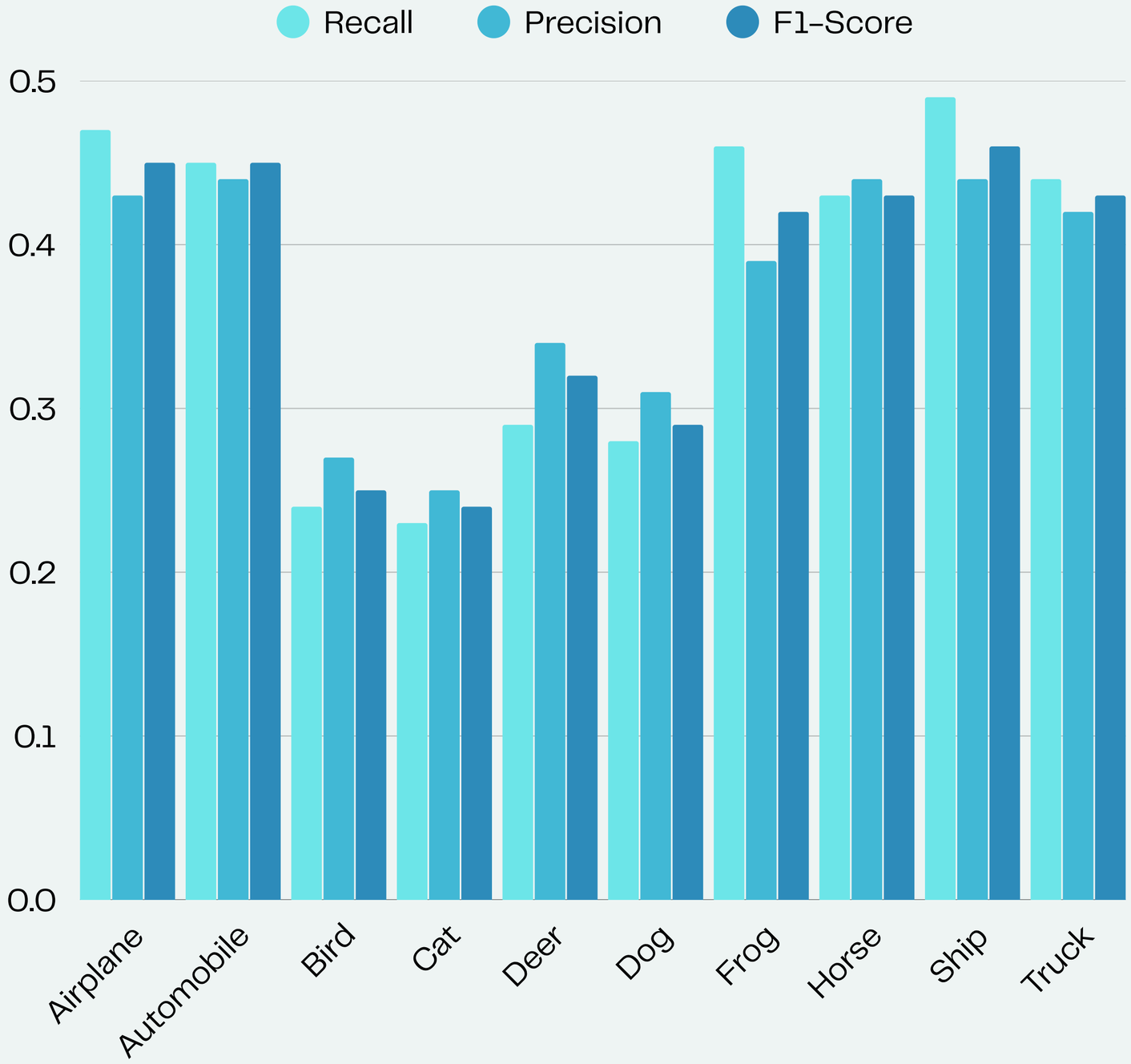
SOFTMAX

```
clf = LogisticRegression(multi_class='multinomial', solver='lbfgs', max_iter=200)
clf.fit(X_pca, y)
```

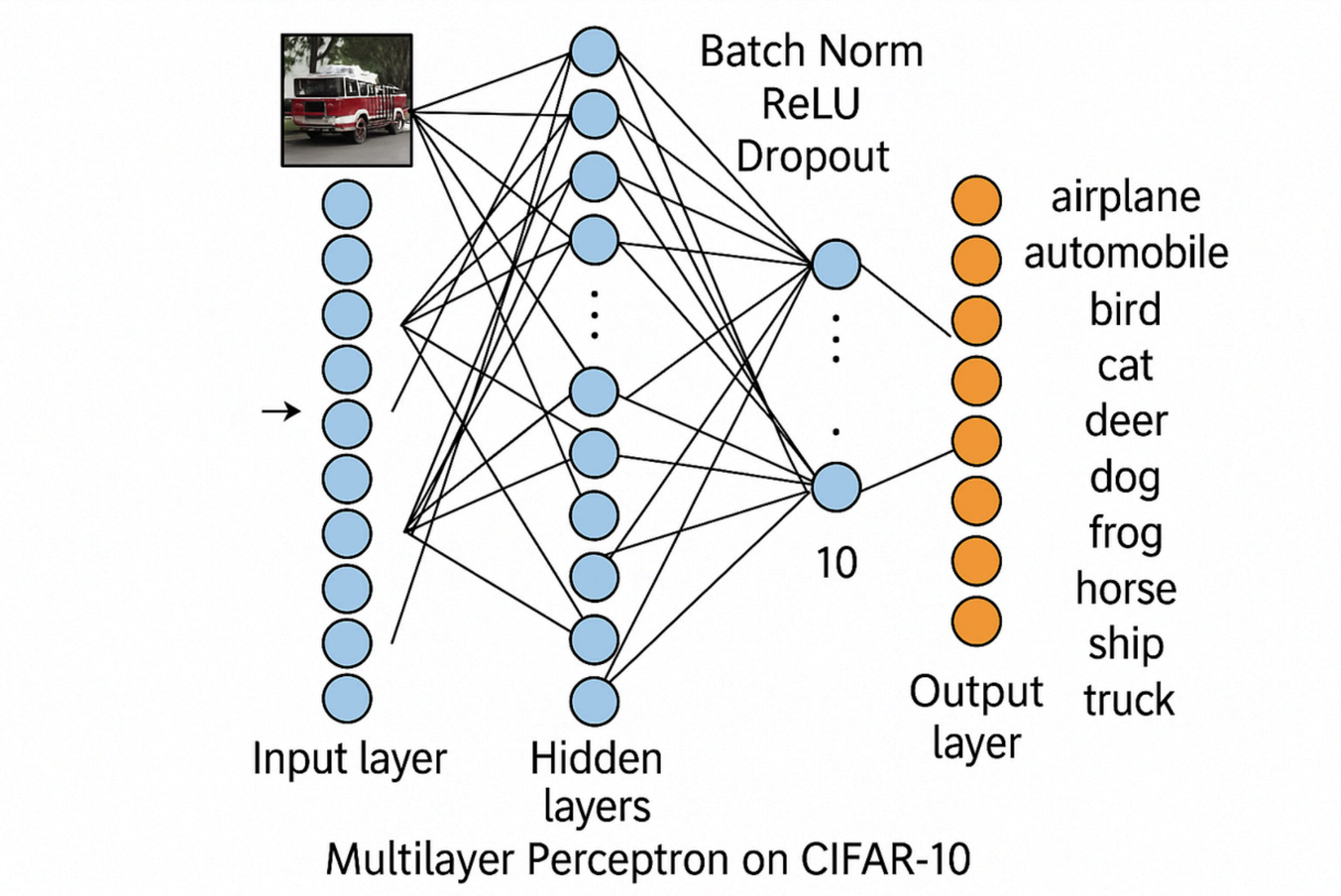


Metric	Score
Accuracy	**0.38**
Macro Avg	**0.37**
Weighted Avg	**0.37**

Class	Precision	Recall	F1-Score	Support
Airplane	0.43	0.47	0.45	1000
Automobile	0.44	0.45	0.45	1000
Bird	0.27	0.24	0.25	1000
Cat	0.25	0.23	0.24	1000
Deer	0.34	0.29	0.32	1000
Dog	0.31	0.28	0.29	1000
Frog	0.39	0.46	0.42	1000
Horse	0.44	0.43	0.43	1000
Ship	0.44	0.49	0.46	1000
Truck	0.42	0.44	0.43	1000



MULTILAYER PERCEPTRON



▶ Training MLP on 500-dim data...
Epoch 1/80 — acc@1k: 52.30%
Epoch 2/80 — acc@1k: 59.10%
Epoch 3/80 — acc@1k: 61.30%
Epoch 4/80 — acc@1k: 67.20%
Epoch 5/80 — acc@1k: 67.50%
Epoch 6/80 — acc@1k: 72.60%
Epoch 7/80 — acc@1k: 74.10%
Epoch 8/80 — acc@1k: 73.70%
Epoch 9/80 — acc@1k: 76.30%

```python
# 5) sample predictions
for i in range(5):
    xi = X_test_p[i:i+1]
    pred = mlp.predict(xi)[0]
    print(f"Sample {i}  true: {y_test[i]}  pred: {pred}")
```

```
Sample 0  true: 3  pred: 3
Sample 1  true: 8  pred: 8
Sample 2  true: 8  pred: 8
Sample 3  true: 0  pred: 8
Sample 4  true: 6  pred: 4
```

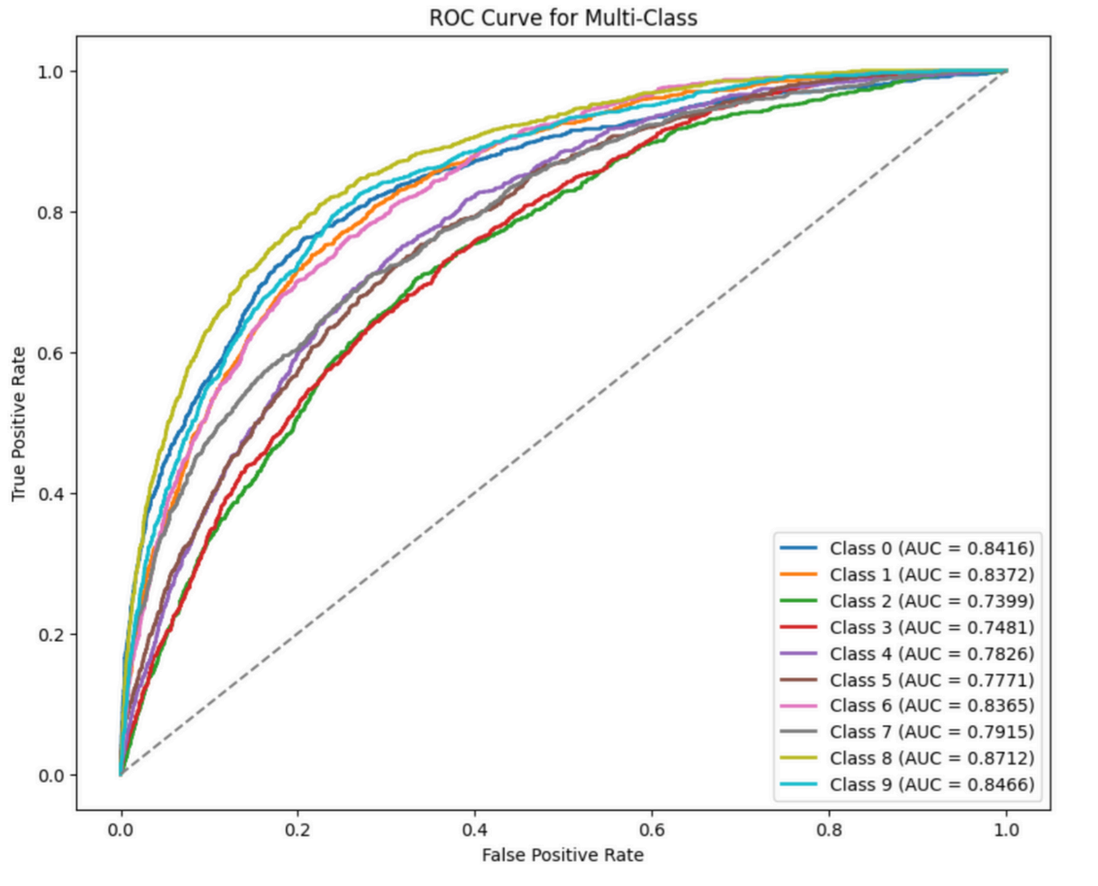
```
▶ Evaluating on test set...
Test accuracy: 46.98%
```

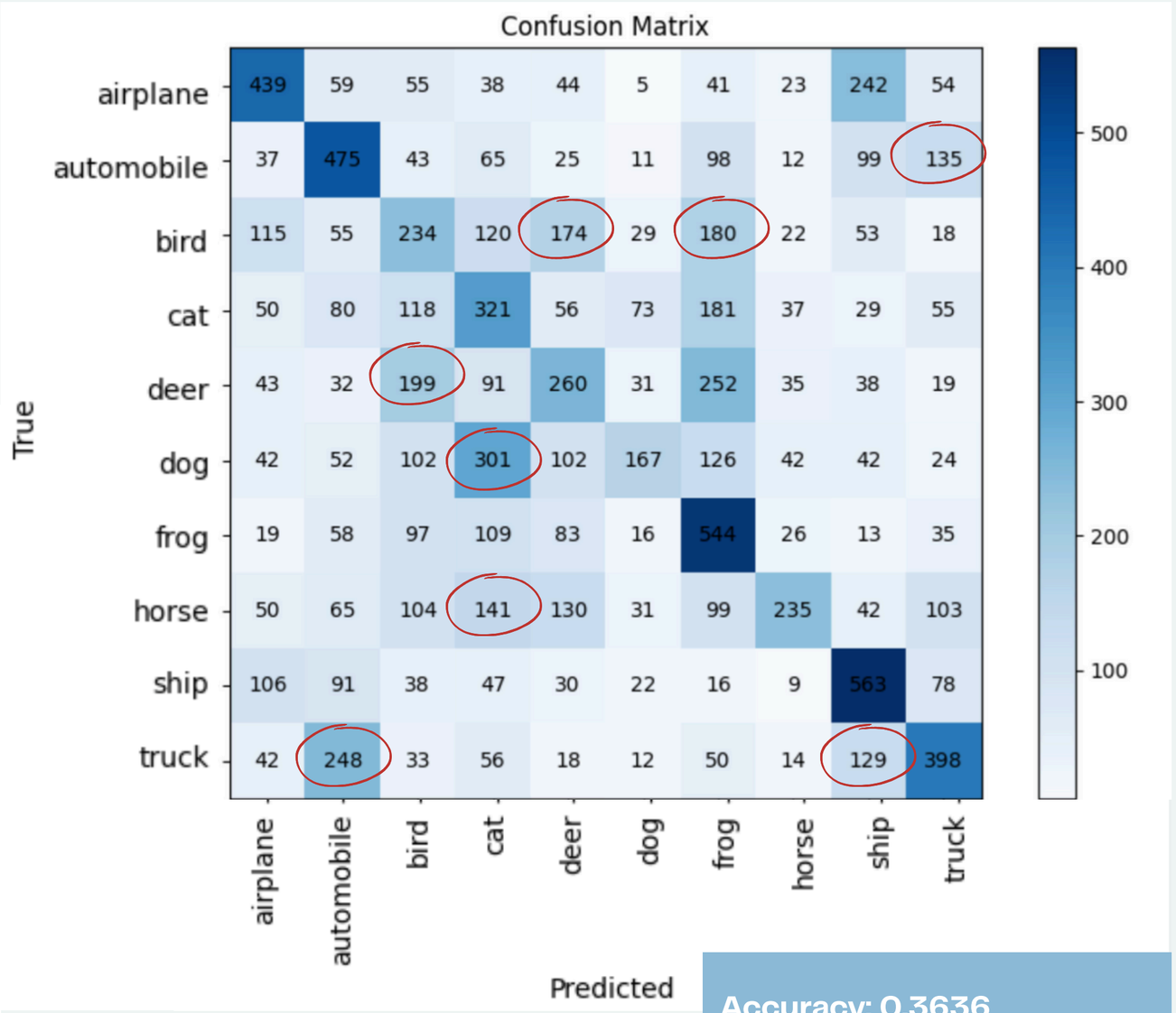
```
class MLP:
    """
    Deep fully-connected Perceptron network with:
      - ReLU activations
      - Softmax output & cross-entropy loss with label smoothing
      - Batch normalization on each hidden layer
      - Dropout on hidden layers
      - L2 weight decay
      - Adam optimizer for all parameters

    Parameters
    ----------
    layer_sizes : list of int
        Sizes of each layer including input and output, e.g. [300, 512, 256, 128, 10].
    lr : float, default=1e-3
        Base learning rate for Adam.
    dropout_rate : float in [0,1), default=0.5
        Drop probability on hidden activations.
    weight_decay : float, default=1e-4
        L2 regularization coefficient.
    label_smoothing : float in [0,1), default=0.1
        ε for smoothing the one-hot targets: ŷ = (1−ε)·y + ε/K.
    beta1, beta2 : floats, default=(0.9, 0.999)
        Adam momentum hyperparameters.
    eps : float, default=1e-8
        Adam & batch-norm stability constant.
    """
```

```
▶ Evaluating on test set...
Test accuracy: 58.56%
```

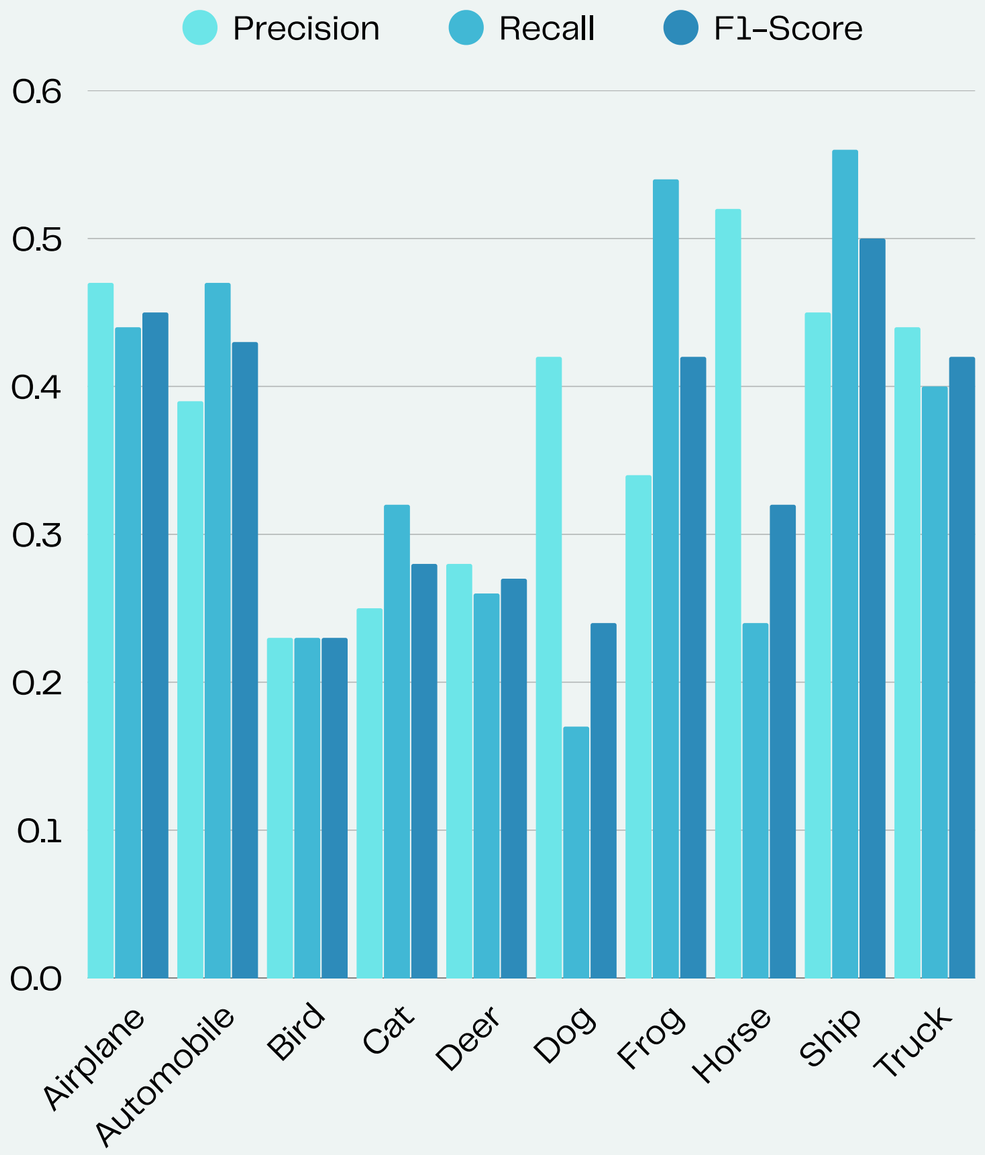
svm



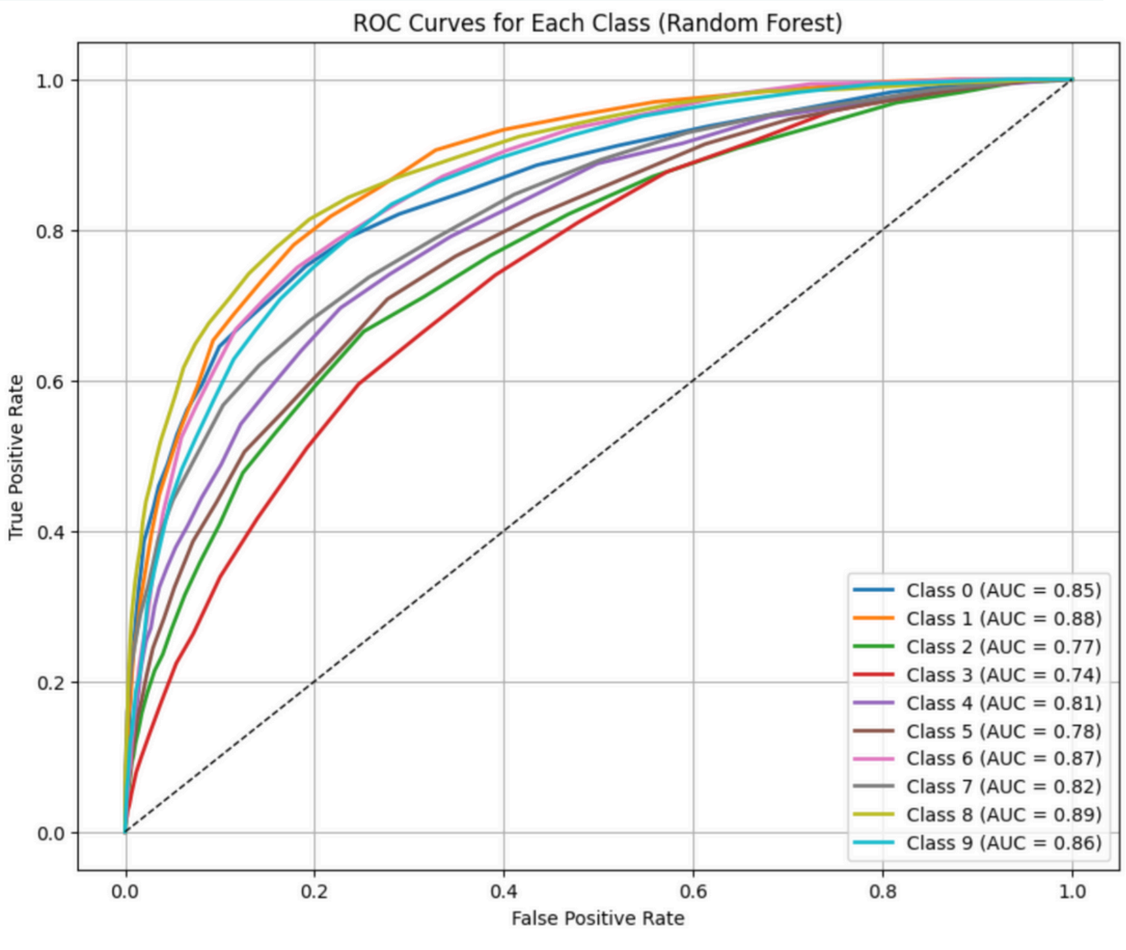


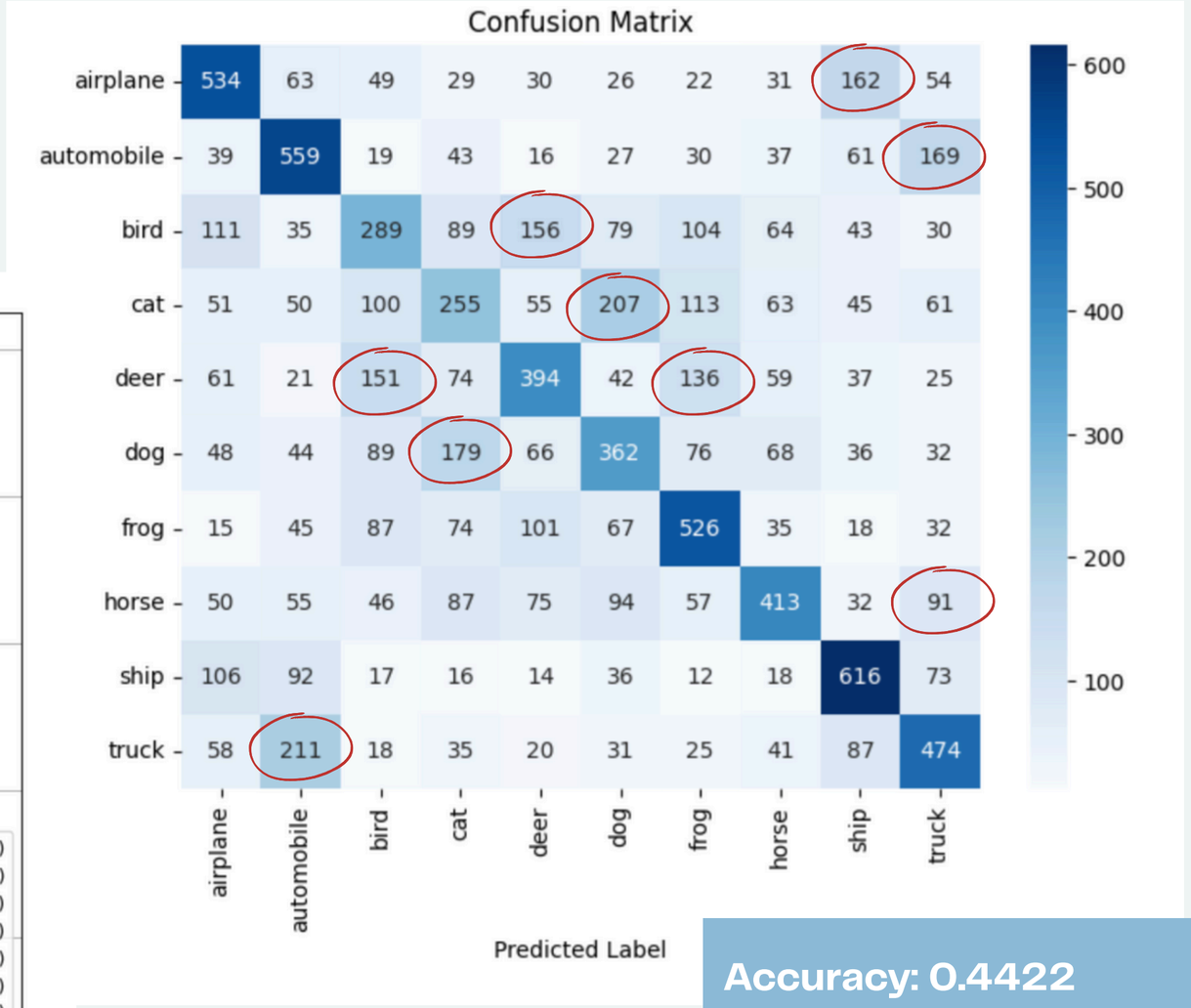
Accuracy: 0.3636
Precision (Weighted): 0.3780
Recall (Weighted): 0.3636
Balanced Accuracy: 0.3636
ROC AUC Score: 0.8072

Class	Precision	Recall	F1-Score	Support
Airplane	0.47	0.44	0.45	1000
Automobile	0.39	0.47	0.43	1000
Bird	0.23	0.23	0.23	1000
Cat	0.25	0.32	0.28	1000
Deer	0.28	0.26	0.27	1000
Dog	0.42	0.17	0.24	1000
Frog	0.34	0.54	0.42	1000
Horse	0.52	0.24	0.32	1000
Ship	0.45	0.56	0.5	1000
Truck	0.44	0.4	0.42	1000



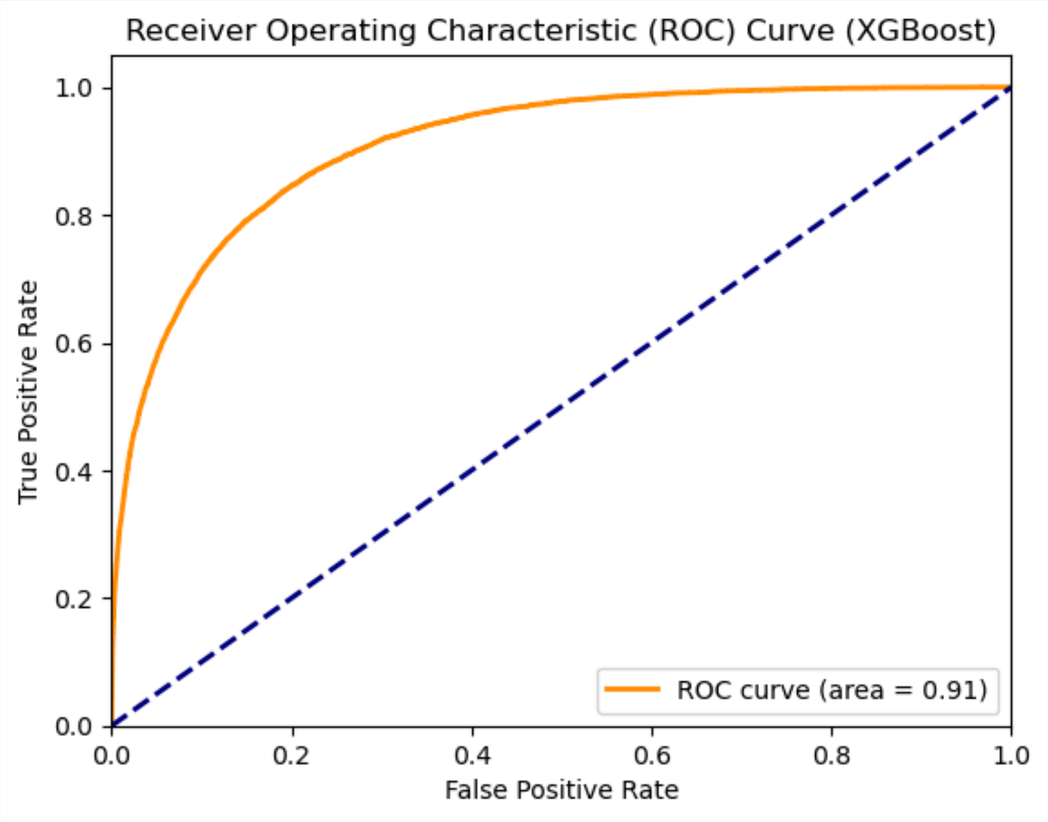
RANDOM FOREST

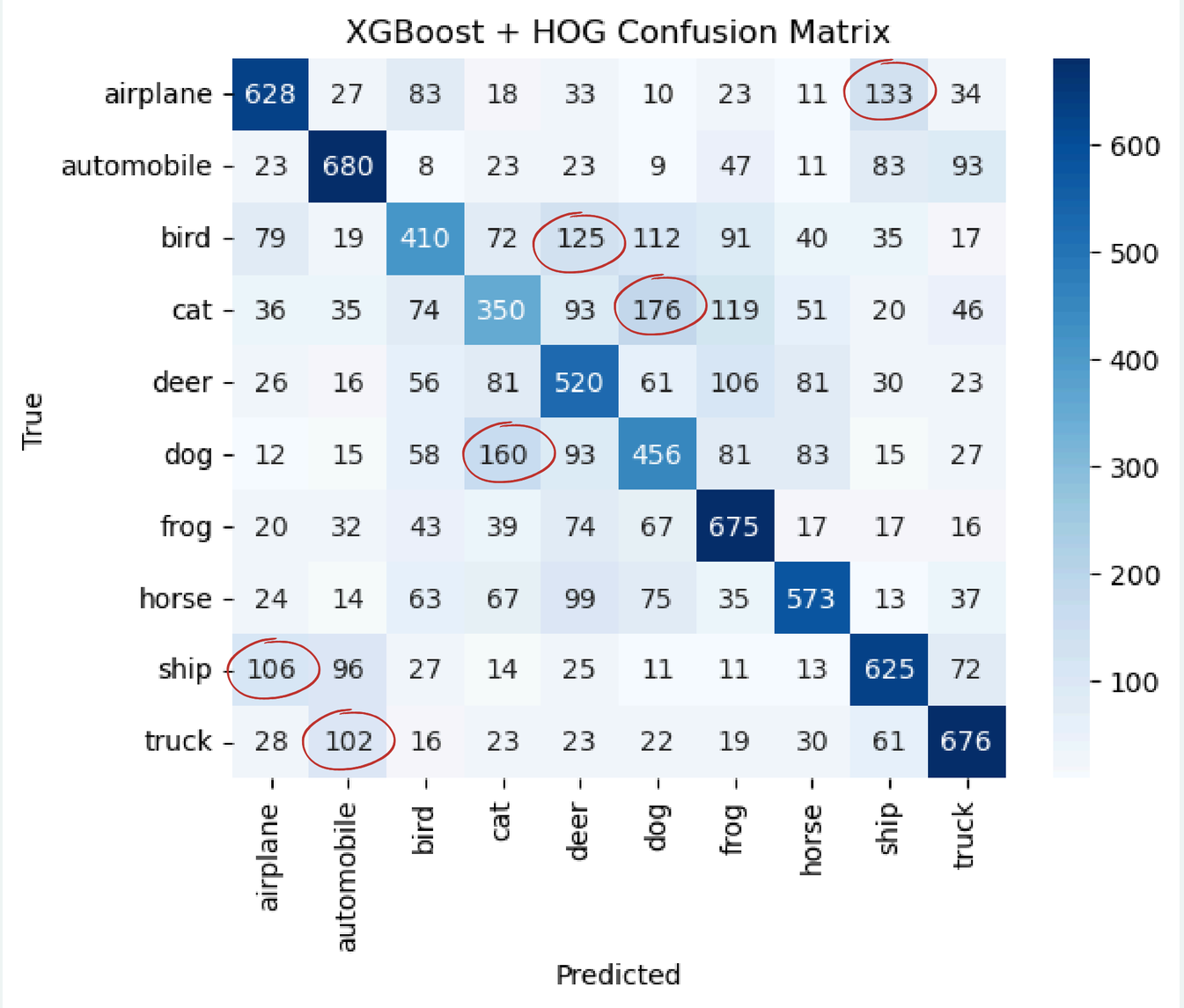




Accuracy: 0.4422
Precision: 0.4368
Recall: 0.4422
Balanced Accuracy: 0.4422
ROC AUC Score: 0.8274

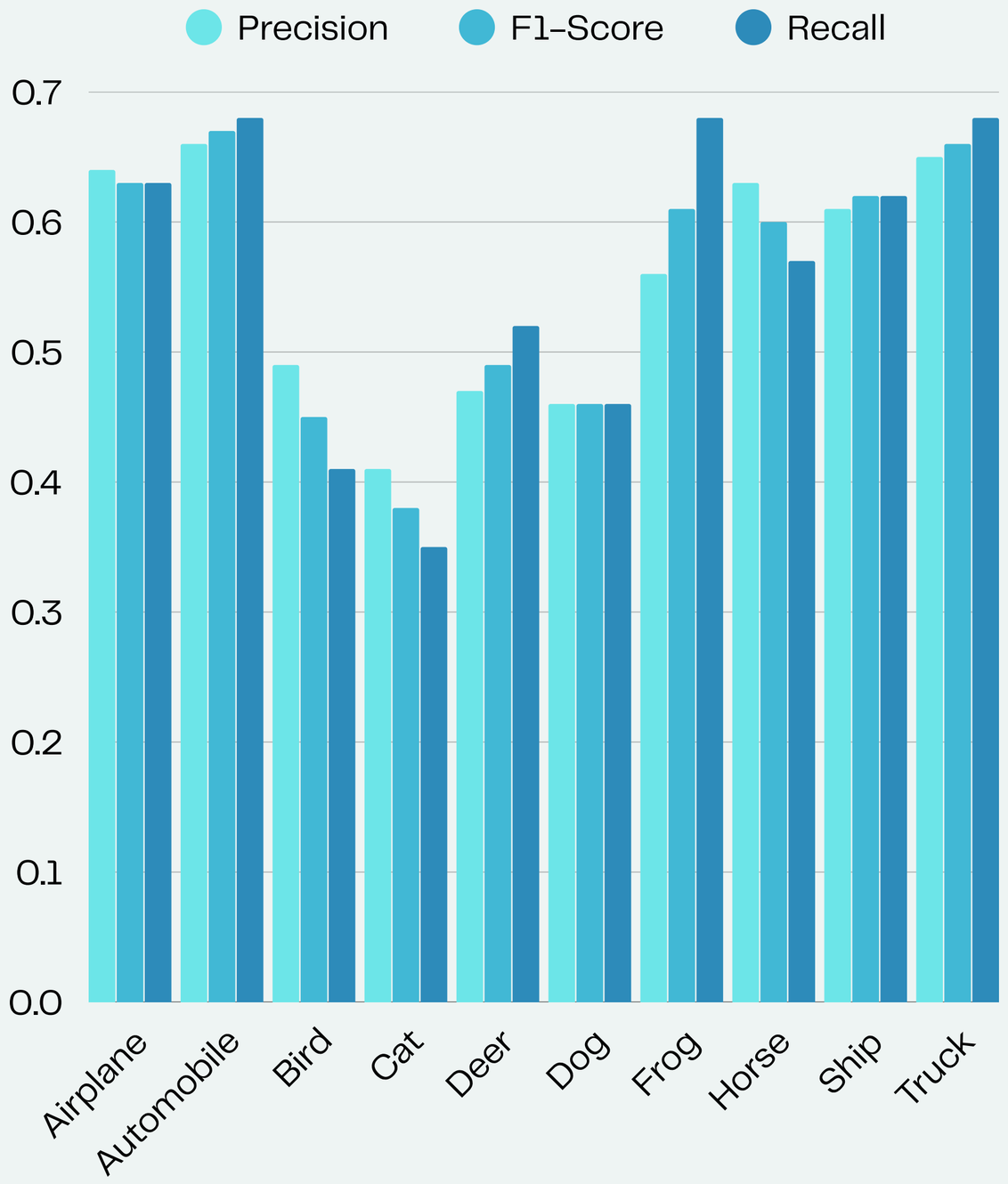
XGBOOST MODEL WITH HOG FEATURE EXTRACTION





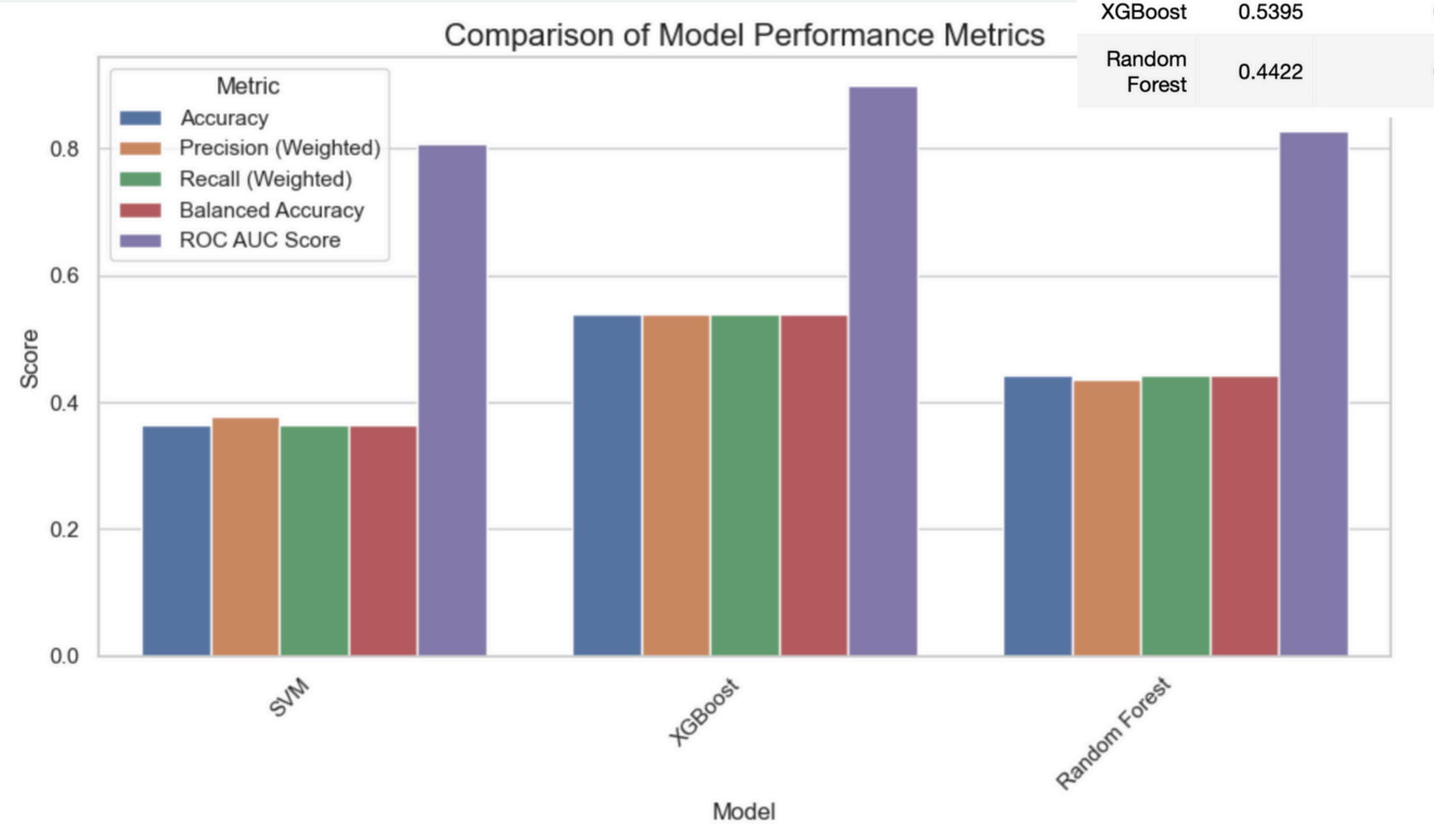
Accuracy: 0.5593
Precision : 0.5568
Recall : 0.5593
Balanced Accuracy : 0.5593
ROC AUC Score : 0.9043

Class	Precision	Recall	F1-Score	Support
Airplane	0.64	0.63	0.63	1000
Automobile	0.66	0.68	0.67	1000
Bird	0.49	0.41	0.45	1000
Cat	0.41	0.35	0.38	1000
Deer	0.47	0.52	0.49	1000
Dog	0.46	0.46	0.46	1000
Frog	0.56	0.68	0.61	1000
Horse	0.63	0.57	0.6	1000
Ship	0.61	0.62	0.62	1000
Truck	0.65	0.68	0.66	1000



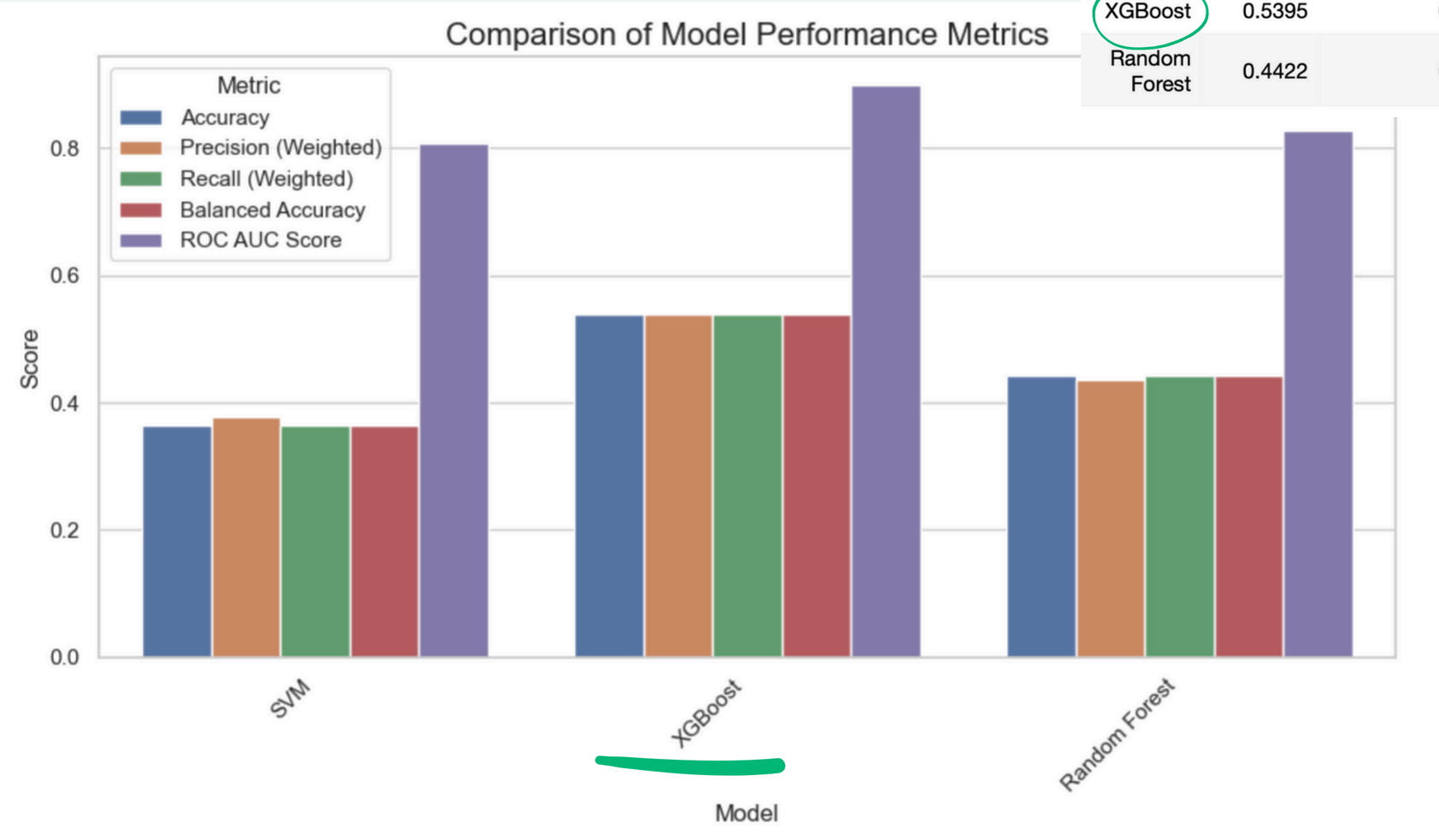
COMPARISON OF THE MODELS

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	Balanced Accuracy	ROC AUC Score
SVM	0.3636	0.377965	0.3636	0.3636	0.807235
XGBoost	0.5395	0.538430	0.5395	0.5395	0.897644
Random Forest	0.4422	0.436785	0.4422	0.4422	0.827437

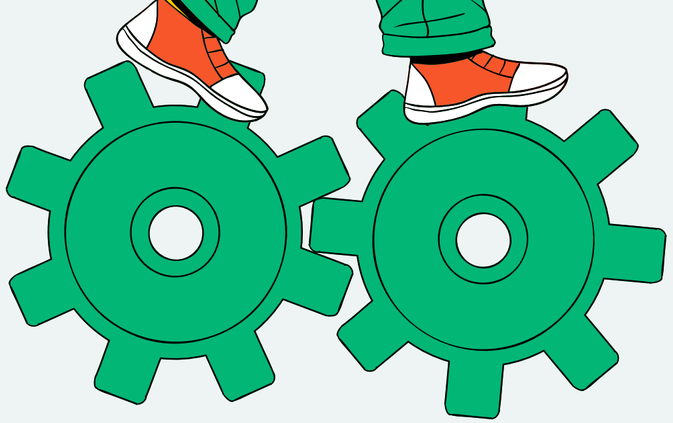


COMPARISON OF THE MODELS

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	Balanced Accuracy	ROC AUC Score
SVM	0.3636	0.377965	0.3636	0.3636	0.807235
XGBoost	0.5395	0.538430	0.5395	0.5395	0.897644
Random Forest	0.4422	0.436785	0.4422	0.4422	0.827437



CLASSIFICATION INTEGRATING K-MEANS CLUSTERING



Patch Sampling & Whitening

Unsupervised Feature Learning

Image Encoding & Spatial Pooling

Linear SVM Classification

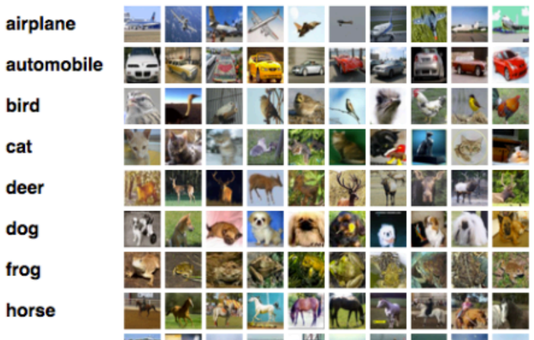
Classifying CIFAR-10 Images Using Unsupervised Feature & Ensemble Learning

Truc Viet "Joe" Le
Heinz College
Carnegie Mellon University
tjle@andrew.cmu.edu

Naassih Gopee
School of Computer Science
Carnegie Mellon University
ngopee@andrew.cmu.edu

ABSTRACT

We perform the image classification task on the CIFAR-10 dataset, where each image belongs to one of the ten distinct classes. The classes are mutually exclusive and are mostly objects and animals. The images are small (32×32 pixels), of uniform size and shape, and RGB coloured. We implement an proposed image preprocessing framework to learn and extract the salient features of the images. The method was demonstrated to increase the classification performance significantly. We testify such claim and see a considerable improvement of more than 15% from the baseline (i.e., without preprocessing). We further experiment with various pa-



Patch Extraction

(6, 6, 3)

Whitening + normalization

Image whitening – de-correlation transformation that transforms each patch into a matrix whose covariance matrix is the identity matrix.

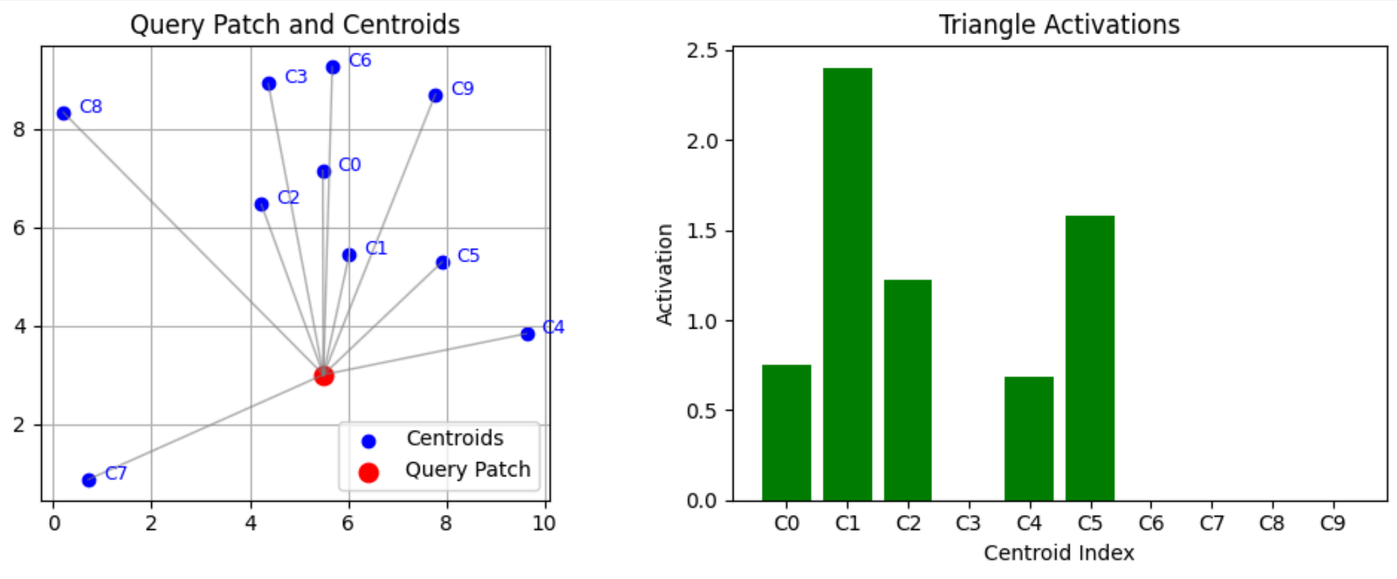
Whitening amplifies variance along decorrelated directions — making patterns and contrasts more pronounced. That’s why the whitened patch may appear more “colorful.”

K-means

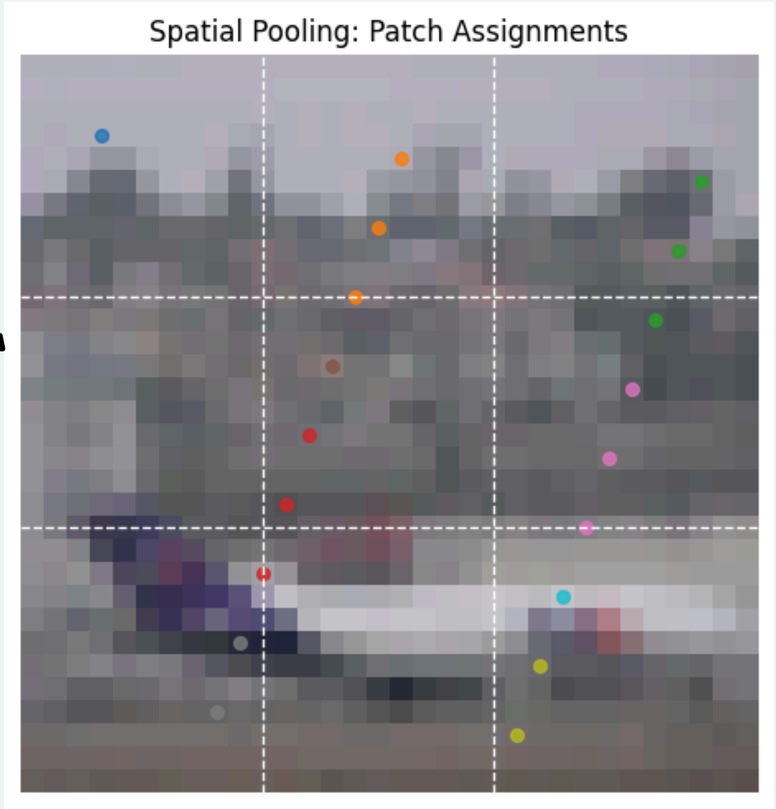
Visualisation of the 400 centroids performed on 200000 random patches

Triangle activation + pooling

Each patch is encoded using triangle activation with respect to the learned centroids.These patch responses are pooled across a 3×3 grid, producing a final K×9–dimensional feature vector.



[0.0, 1.74, 0.0, 0.52, 0.0, ..., 0.0], shape for one image: (676, K)

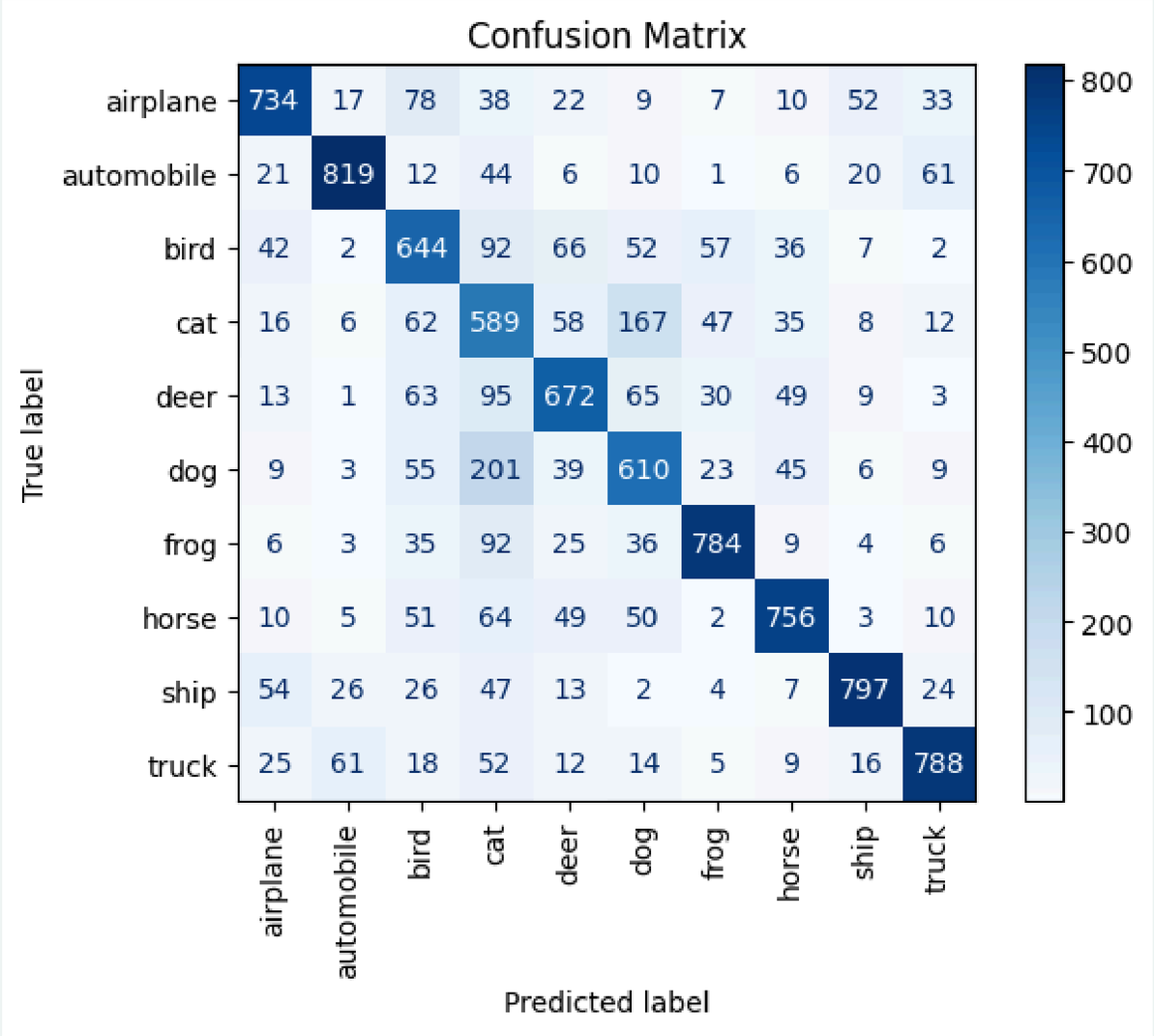


Each image: (9xK)

Linear SVM

K = 1600

Number of training samples: 50000
Number of test samples: 10000



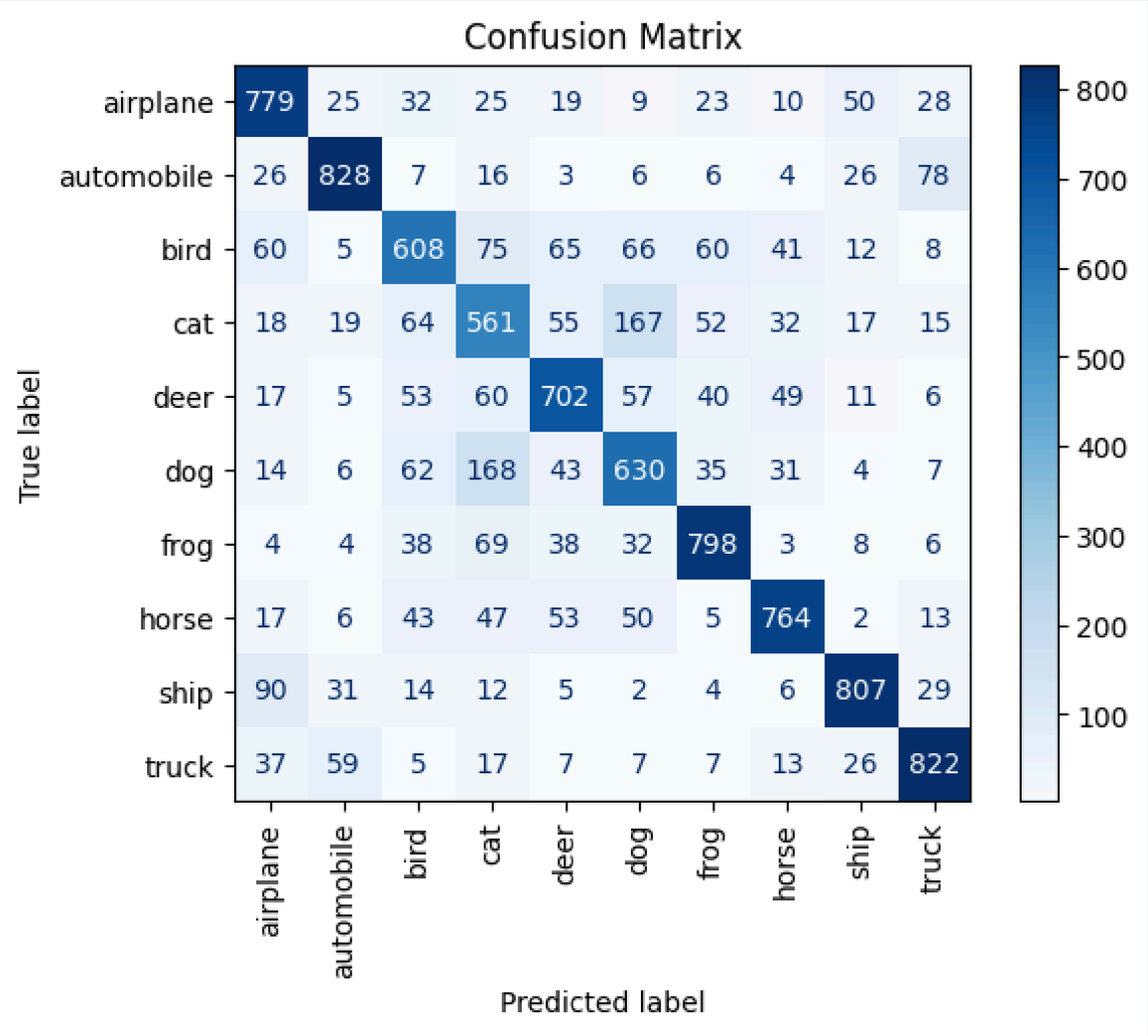
Final Accuracy: 71.93%

automobile: 81.90%
ship: 79.70%
truck: 78.80%
frog: 78.40%
horse: 75.60%
airplane : 73.40%
deer: 67.20%
bird: 64.40%
dog: 61.00%
cat: 58.90%

Linear SVM

K = 4000

Number of training samples: 35000
Number of test samples: 10000

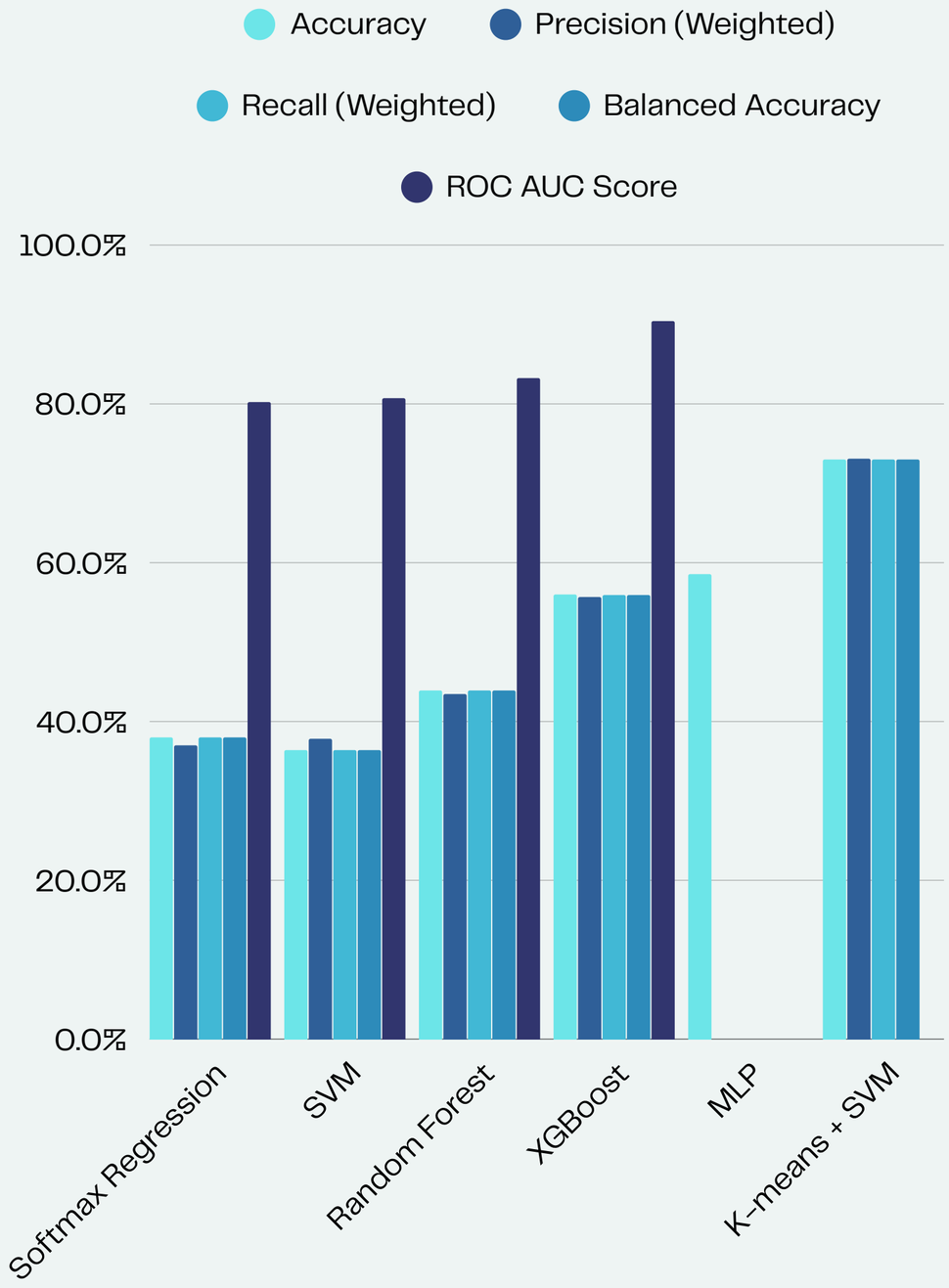


Final Accuracy 72.99%

automobile: 82.80%
truck: 82.20%
ship: 80.70%
frog: 79.80%
airplane: 77.90%
horse: 76.40%
deer: 70.20%
dog: 63.00%
bird: 60.80%
cat: 56.10%

OVERALL METRICS

Model	Accuracy	Precision (Weighted)	Recall (Weighted)	Balanced Accuracy	ROC AUC Score
Softmax Regression	38.00%	37.00%	38.00%	38.00%	80.23%
SVM	36.40%	37.83%	36.40%	36.40%	80.72%
Random Forest	43.91%	43.46%	43.91%	43.91%	83.25%
XGBoost	56.00%	55.68%	55.93%	55.93%	90.43%
MLP	58.56%	—	—	—	—
K-means + SVM	73.00%	73.10%	73.00%	73.00%	—



CONCLUSION

Based on the evaluation of multiple models, the **K-means + SVM** pipeline emerged as the most effective approach. While traditional classifiers like **Softmax Regression, SVM, Random Forest**, and **XGBoost** achieved moderate ROC AUC scores, their accuracy remained below satisfactory levels, with the best among them (XGBoost) reaching only 55.93% accuracy. The **MLP** model performed slightly better with 58.56% accuracy, but still fell short of expectations. In contrast, the K-means + SVM combination achieved a significantly higher accuracy of 73.00%, along with strong precision and recall scores, indicating consistent and reliable classification performance

REFERENCES

Le, T. V., & Gopee, N. (n.d.). Classifying CIFAR-10 Images Using Unsupervised Feature & Ensemble Learning. Carnegie Mellon University. Retrieved from https://www.andrew.cmu.edu/user/tjle/

Coates, A., Lee, H., & Ng, A. Y. (2011). An analysis of single-layer networks in unsupervised feature learning. In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS), 2011.

Dalal, N., & Triggs, B. (2005). Histograms of Oriented Gradients for Human Detection.In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), Vol. 1, pp. 886-893.