CSE343: Machine Learning

Assignment-2

Himanshu Raj (2022216)

September 13, 2024

Section C

The problem is implementing various classification models using the scikit-learn package. The data analysis and model training can be found in the notebook.

1) Extensive Data Analysis

a) The dataset contains 840 images per class - 12,600 images for all 15 classes. The mode value of width is 275px, and height is 183px. The mean value of width is 260px, and height is 196px. The mode value of the aspect ratio (width/height) is 1.5.

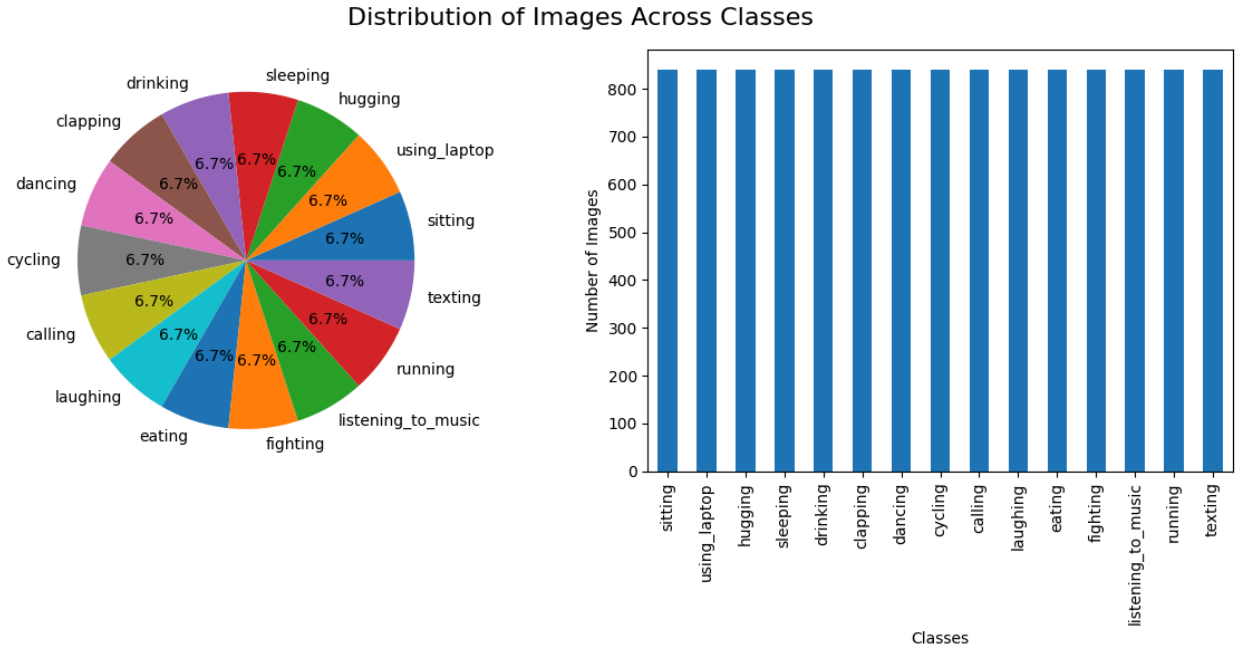
```
label
sitting               840
using_laptop          840
hugging               840
sleeping              840
drinking              840
clapping              840
dancing               840
cycling               840
calling               840
laughing              840
eating                840
fighting              840
listening_to_music    840
running               840
texting               840
Name: count, dtype: int64
```

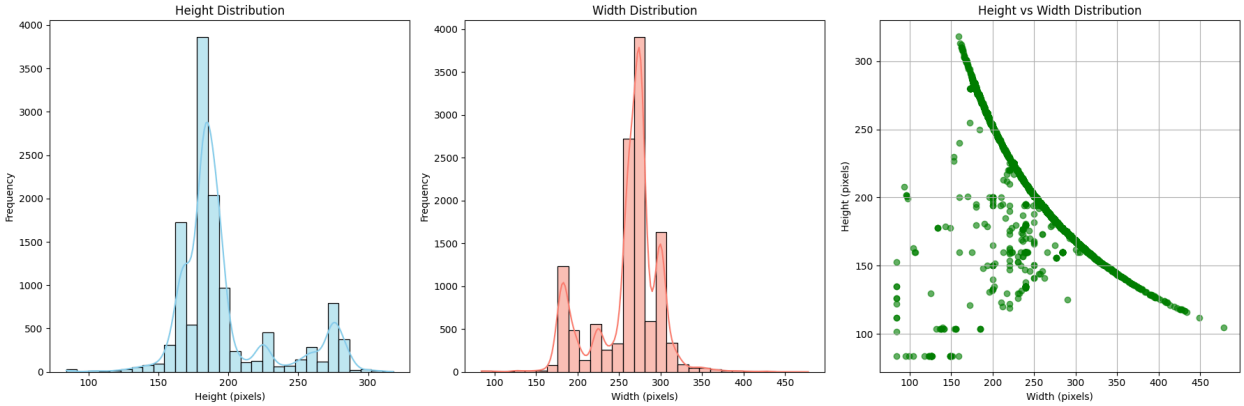
```
width
275    3074
300    1038
262     746
183     703
259     646
       ...
161       1
385       1
478       1
379       1
419       1
Name: count, Length: 269, dtype: int64
height
183    3310
168    1328
194     705
275     644
192     617
       ...
293       1
116       1
102       1
318       1
312       1
Name: count, Length: 199, dtype: int64
```

```
aspectratio
1.50    3111
1.79    1054
1.36     749
1.34     696
0.67     678
        ...
3.11       1
2.85       1
2.77       1
2.92       1
3.49       1
Name: count, Length: 233, dtype: int64
```

```
Mean value for width is  260.38103174603174
Mean value for height is  196.57357142857143
```

b) These graphs show that all classes have an equal number (840) of samples. The mode value of width is 275px, and height is 183px.





Distribution of image sizes (width and height)

sitting

sitting

sitting

sitting

using_laptop

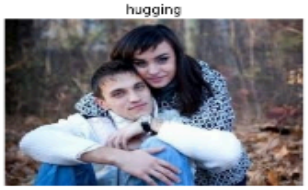
using_laptop

using_laptop

using_laptop

hugging

hugging



hugging

hugging

sleeping

sleeping

sleeping

sleeping

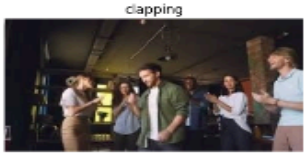
drinking

drinking

drinking

drinking

clapping



clapping

clapping

clapping

dancing

dancing

dancing

dancing

cycling

cycling

cycling

cycling

calling

calling

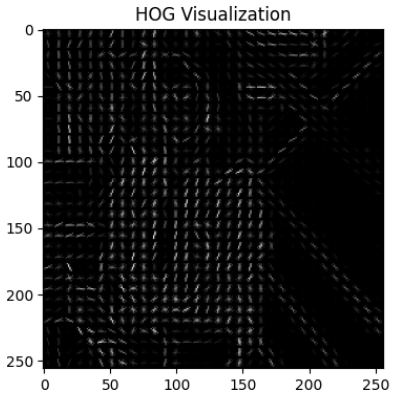
calling

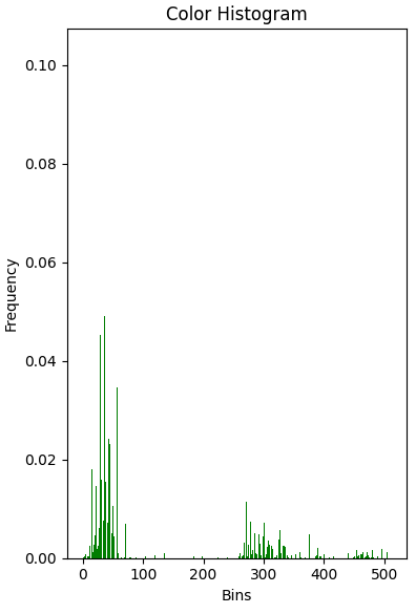
calling

c) This dataset has no class imbalances, the initial probability is the same for all classes.
If imbalances were present, we could have down-sampled data to the minimum amount present in any class; or up-sampled using image augmentation in classes having lesser samples to a number that will be the same for all classes (usually the maximum number of samples in any class); or we could use a combination of both to a number that will ensure equal samples in each class.

2) Feature Extraction

I have extracted 5 features from the images in the dataset: Histogram of Oriented Gradients (object detection), Color Histograms (distribution of colors), Local Binary Patterns (texture detection), Color Moments (distribution of colors), and Image Gradients with Sobel filter (edge detection).
I used the image dimension as 64x64 and grayscaled image to extract HOG, LBP, and Image Gradients. I used the image in HSV color space to extract Color Histograms and Color Moments.
Feature and size mapping is HOG- 1764, Color Histograms- 512, LBP- 18, Color Moments- 9, and Sobel Gradients- 4096, making a total of 6399 features.
Below is a visualization of HOG features and Color Histograms.





3) Model Selection

We combine all the features extracted above. We split the dataset into train:test (80:20), then apply standard scaling, then PCA to reduce to 500 features. We train Naive Bayes, Decision Tree, Random Forests and Perceptron models with default parameters on this split. We are using random_state=42 wherever applicable.

```
Naive Bayes Accuracy: 0.17
Decision Tree Accuracy: 0.13
Perceptron Accuracy: 0.21
Random Forest Accuracy: 0.20
```

We see that the decision tree model has the lowest accuracy, but the random forest model has close to the highest among them, also it can be tweaked to improve accuracy further. The 3 models except Random Forest take comparatively less time to train and don't have much scope for improvement. We will move forward with the Random Forest model and perform parameter tuning for better accuracy.

4) **Best Model**

After training RandomForestClassifier on various parameters and under various conditions, I observed that applying PCA to reduce feature dimensions leads to less accuracy. Considering color moments and Sobel gradients as features also leads to lower accuracy.

Accuracy maxes out at a max_depth of 30 with n_estimators as 500 with a value of 33.9. This model takes almost 2 minutes to train, which is more than any model from part c).

```
model = RandomForestClassifier(n_estimators=500, max_depth=30, random_state=42, n_jobs=-1)
model.fit(X_train_s, y_train)
y_pred = model.predict(X_test_s)
acc = accuracy_score(y_pred, y_test)
print(acc)
```

```
0.3388888888888889
```

After reaching this maximum accuracy after many attempts, I applied XG boosting algorithm on Random Forest using XGBClassifier to improve accuracy even further. Applying PCA and considering color moments as features leads to lower accuracy. This model takes ~3 hours to train but provides the best accuracy of all classifiers. This provides a maximum accuracy of 38.9 with parameters n_estimators=500, and learning_rate=0.1.

```
#load trained model
xgb_model = pickle.load(open('model.pkl', 'rb'))
y_pred = xgb_model.predict(X_test_s)
acc = accuracy_score(y_pred, y_test)
print(acc)
# 0.3884920634920635
```

```
0.3884920634920635
```

Himanshu Raj (2022216)

Date : / /
Page No.

ML Asgn2 Sec A

a) To find - $P(D \mid 4\%\ \text{profit})$

given - $P(D) = 0.8$

D - issued dividend
$\bar{D}$ - didn't issue divident.

company issuing dividends, $\mu = 10$, $\sigma^2 = 36$, $\sigma = 6$
company not issuing dividends, $\mu = 0$, $\sigma^2 = 36$, $\sigma = 6$

$$P(D \mid 4\%\ pr) = \frac{P(4\%\ \text{profit} \mid D) \times P(D)}{P(4\%\ \text{profit})}$$

$$P(4\%\ \text{profit}) = P(4\%\ pr \mid D)\, P(D) + P(4\%\ pr \mid \bar{D})\, P(\bar{D})$$

$$P(4\%\ pr \mid D) = \frac{1}{6\sqrt{2\pi}} e^{-\frac{(4-10)^2}{2\times 36}} \qquad \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$= \frac{1}{6\sqrt{2\pi}} e^{-1/2} \simeq 0.040$$

$$P(4\%\ pr \mid \bar{D}) = \frac{1}{6\sqrt{2\pi}} e^{\frac{-(4-0)^2}{2\times 36}} = \frac{1}{6\sqrt{2\pi}} e^{-2/9} \simeq 0.0353$$

$$P(4\%\ \text{profit}) = \frac{1}{6\sqrt{2\pi}} \left(0.8\, e^{-1/2} + 0.2\, e^{-2/9}\right)$$

$$P(D \mid 4\%\ \text{profit}) = \frac{\frac{1}{6\sqrt{2\pi}} e^{-1/2} \times 0.8}{\frac{0.2}{6\sqrt{2\pi}} \left(4e^{-1/2} + e^{-2/9}\right)} = \frac{4e^{-1/2}}{4e^{-1/2} + e^{-2/9}}$$

$$= 0.7518$$

There is an 75.18 % chance that a company will issue dividends if it had 4% profit increase last year.

Date : / /
Page No.

b Target – Y = Attended Class

$H(Y) = -\frac{7}{12}\log_2\frac{7}{12} - \frac{5}{12}\log_2\frac{5}{12}$ [$-P(yes)\log_2 P(yes) - P(No)\log_2 P(No)$]

$= (-0.583)(-0.777) - 0.4816(-1.263)$

$= 0.453 + 0.526 = \underline{0.979}$

IG(Had Proper sleep) $= H(Y) - H(Y|X)$

$= 0.979 - \left[\frac{6}{12}\left(\frac{6}{6}\log_2 1 + 0\log_2 0\right) - \frac{6}{12}\left(\frac{1}{6}\log_2\frac{1}{6} + \frac{5}{6}\log_2\frac{5}{6}\right)\right]$

$0 \leftarrow$

$= 0.979 + \frac{6}{12}(-0.430 - 0.219)$

$= 0.979 - 0.3245 = \underline{0.6548}$

IG (Class Time)

$= 0.979 - \left[\frac{-4}{12}\left(\frac{3}{4}\log_2\frac{3}{4} + \frac{1}{4}\log_2\frac{1}{4}\right) - \frac{4}{12}\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{4}\log_2\frac{2}{4}\right)\right.$

$\left. - \frac{4}{12}\left(\frac{2}{4}\log_2\frac{2}{4} + \frac{2}{2}\log_2\frac{2}{4}\right)\right]$

$= 0.979 + \frac{4}{12}(-0.811 - 1 - 1)$

$= 0.979 - 0.937 = \underline{0.042}$

IG (weather)

$= 0.979 - \left[\frac{-5}{12}\left(\frac{4}{5}\log_2\frac{4}{5} + \frac{1}{5}\log_2\frac{1}{5}\right) - \frac{2}{12}\left(\frac{2}{2}\log_2 1\right)\right.$ 0

$\left. - \frac{5}{12}\left(\frac{3}{5}\log_2\frac{3}{5} + \frac{2}{5}\log_2\frac{2}{5}\right)\right]$

$= 0.979 + \frac{5}{12}(-0.722 - 0.971)$

$= 0.979 - 0.705 = \underline{0.274}$

Date : / /
Page No.

∵ IG (had proper sleep) is highest and hence it is the root node.

If had proper sleep = yes, then tree terminates and if had proper sleep = no.

$$H(\text{no proper sleep}) = -\frac{1}{6}\log\frac{1}{6} - \frac{5}{6}\log\frac{5}{6}$$

$$= 0.431 + 0.220 = \underline{0.651}$$

$$H(\text{ML} \mid \text{temp}) = -\frac{2}{6}\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right) - 2\left(\frac{2}{6}\left(\cancel{1\log 1}^{\to 0} + \cancel{0\log 0}\right)\right)$$

given no sleep

$$= 0.333$$

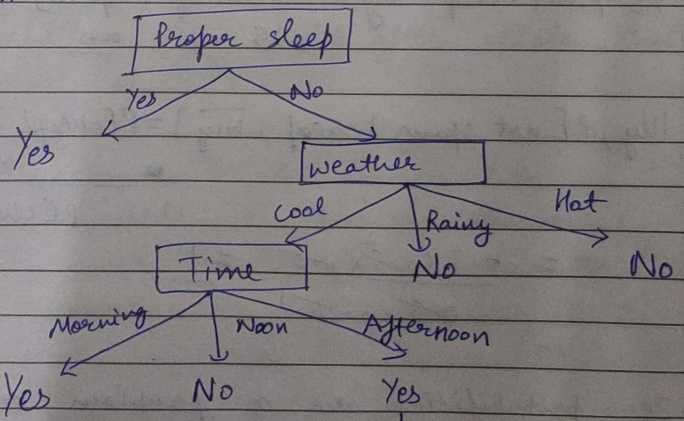
$$H(\text{ML} \mid \text{weather}) = -\frac{2}{6}\left(\cancel{0\log 0 + 1\log 1}^{\to 0}\right)\times 2 - \frac{2}{6}\left(\frac{1}{2}\log\frac{1}{2} + \frac{1}{2}\log\frac{1}{2}\right)$$

given no sleep

$$= 0.333$$

We can ~~split on either~~ temp or weather

The decision tree -



→ No data given for tuple (Afternoon, cool, No) so its an assumption. (can be no as well)

Date : / /
Page No.

d a) $P[\text{spam}] = \frac{1}{2} = 0.5$

$P[\text{Not spam}] = \frac{1}{2} = 0.5$

$P(\text{buy}=1 \mid \text{spam}) = 1$ $\quad$ $P(\text{buy}=1 \mid \text{not spam}) = 1/2$

$P(\text{buy}=0 \mid \text{spam}) = 0$ $\quad$ $P(\text{buy}=0 \mid \text{not spam}) = 1/2$

$P(\text{cheap}=1 \mid \text{spam}) = 1/2$ $\quad$ $P(\text{cheap}=1 \mid \text{not spam}) = 1/2$

$P(\text{cheap}=0 \mid \text{spam}) = 1/2$ $\quad$ $P(\text{cheap}=0 \mid \text{not spam}) = 1/2$

$P(\text{buy}) = \frac{3}{4}$ $\quad$ $P(\text{cheap}) = \frac{1}{2}$

b) $P[\text{spam} \mid \text{cheap}, \overline{\text{buy}}] = \frac{P[\text{cheap}, \overline{\text{buy}} \mid \text{spam}] \times P[\text{spam}]}{P[\text{cheap}, \overline{\text{buy}}]}$

from our naive assumption,
we assume the probabilities of features to be independent.

$P[\text{spam} \mid \text{cheap}, \overline{\text{buy}}) = P[\text{cheap} \mid \text{spam}] \cdot P[\overline{\text{buy}} \mid \text{spam}]$

$= 0.5 \times 0 = 0$

Illy, $P[\text{not spam} \mid \text{cheap}, \overline{\text{buy}}] = \frac{P[\text{cheap} \mid \overline{\text{spam}}] \cdot P[\overline{\text{buy}} \mid \overline{\text{spam}}] \times P[\text{not spam}]}{P[\text{cheap}] P[\overline{\text{buy}}]}$

$= \frac{0.5 \times 0.5^2 \times 0.5}{0.5 \times 0.25} = 1$

c) Zero probabilities are a problem because Naive Bayes assumes features to be independent and multiplies their independent probabilities together and in most cases data is not enough to cover all such probabilities and they are estimated to 0 and thus product is also zero.

Date : / /
Page No.

In this case, $P[\overline{buy} \mid spam] = 0$ which affected $P[cheap, \overline{buy} \mid spam]$.

To avoid zero probabilities, we can use laplace smoothening or m-estimation.

$$\text{Laplace } (A_i \mid C) = \frac{N_{ic}+1}{N_c + C}$$ (if $N_{ic}=0$, the prob will be non-zero, $\frac{1}{N_c+C}$)

$$\text{m-estimation} = \frac{N_{ic} + mp}{N_c + m}$$ (if $N_{ic}=0$, prob will be non-zero $\frac{mp}{N_c+m}$)

c) γ - min margin across all samples

~~margin mistake : occurs~~

To prove - no. of updates $\leq \frac{8}{\gamma^2}$ and the perceptron algo halts after update that is a polynomial of $1/\gamma$.

Proof - Let $w_{k+1} = x_0 + x_1 + x_2 + \dots + x_k$ — ①

Since the data is linearly separable, $\exists$ an w_0 s.t. $w_0^T x_n > 0$ for $x_1, x_2 \dots x_n \in C_1$ (class)

$$\frac{\gamma}{2} = \min_{x(n) \in C_1} w_0^T x(n) \quad \text{(min margin)}$$

$\Downarrow$

k updates are done, that means the margin was greater than $\frac{\gamma}{2}$.

Date : / /
Page No.

Multiplying w_o^T on both sides in ①

$$w_o^T w_{k+1} = w_o^T x_0 + w_o^T x_1 + \dots + w_o^T x_k$$

lower bound — $w_o^T w_{k+1} \geq \frac{k\gamma}{2}$ (k updates already done and min margin = $\frac{\gamma}{2}$)

$$w_o^T w_{k+1} = w_o^T x_0 + w_o^T x_1 + \dots + w_o^T x_i > \frac{k\gamma}{2}$$

Assumption - Euclidean length = 1

$\|w_o\| = 0$

$$\|w_k\|^2 = \underbrace{\|w_o\|^2}_{0} + \sum_{i=1}^{k} \underbrace{\|w_o^T x_i\|^2}_{1}$$

$$\|w_k\|^2 = k$$

$\|w_k\| \leq \sqrt{k}$ — upper bound

combining bounds

$$\underbrace{w_k \|w_o^T\|}_{\frac{k\gamma}{2}} \leq \|w_k\| \|w_o^T\| \leq \underbrace{\sqrt{k}\, \|w_o^T\|}_{\sqrt{k}}$$

$\|w_o^T\| = 1$ (assumption)

$$\frac{k\gamma}{2} \leq \sqrt{k}$$

$$k \leq \frac{4}{\gamma^2} \quad \text{(loose bound)}$$

this is excluding margin mistakes.

now, accounting for margin mistakes, the bound can be tightened to -

$$k \leq \frac{8}{\gamma^2}$$

Date : / /
Page No.

now, margin threshold is $(1-\varepsilon)\gamma$

for this, mistake bound will be -

$$w_k \|w_0^T\| \leq \|w_k\| \|w_0^T\| \leq \sqrt{k} \|w_0^T\|$$

$w_k \|w_0^T\| \to k(1-\varepsilon)\gamma$

$\|w_0^T\| \to \sqrt{k}$

$$\therefore k(1-\varepsilon)\gamma \leq \sqrt{k}$$

$$k \leq \frac{1}{(1-\varepsilon)^2\gamma^2}$$

again, we account for margin mistakes and tighten the bound to

$$k \leq \frac{8}{(1-\varepsilon)^2\gamma^2}$$