CSE343: Machine Learning
Assignment-2

Shubham Sharma (2021099)

October 1, 2024

1 Section A (Theoretical)

Solution (a)

Given Values

- $P(D) = 0.8$: Probability of issuing a dividend.
- $P(ND) = 0.2$: Probability of not issuing a dividend.
- Profit increase for dividend-issuing companies $\sim \mathcal{N}(10\%, 36\%)$.
- Profit increase for non-dividend-issuing companies $\sim \mathcal{N}(0\%, 36\%)$.
- Observed profit increase: $P = 4\%$.

We want to calculate $P(D|P)$, the probability of issuing a dividend given a 4% profit increase. By Bayes' Theorem:

$$P(D|P) = \frac{P(P|D)P(D)}{P(P|D)P(D) + P(P|ND)P(ND)}$$

Step 1: Likelihood of Profit Increase

Both the dividend and non-dividend cases follow normal distributions; we can use the normal distribution to calculate the likelihoods.

$$P(P|D) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(4-10)^2}{2\sigma^2}}$$

$$P(P|ND) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(4-0)^2}{2\sigma^2}}$$

Since the standard deviation is the same for both, the normalization constant cancels out, so we only need to calculate the exponentials.

For $P(P|D)$:

$$P(P|D) = e^{-\frac{(4-10)^2}{2\times 36^2}} = e^{-0.01389} \approx 0.9862$$

For $P(P|ND)$:

$$P(P|ND) = e^{-\frac{(4-0)^2}{2\times 36^2}} = e^{-0.00617} \approx 0.9938$$

Step 2: Apply Bayes' Theorem

Substitute the values into Bayes' Theorem:

$$P(D|P) = \frac{0.9862 \times 0.8}{0.9862 \times 0.8 + 0.9938 \times 0.2} = \frac{0.78896}{0.78896 + 0.19876} = \frac{0.78896}{0.98772} \approx 0.798$$

The likelihood that a company with a 4% profit increase will issue a dividend is approximately **79**%.

Solution (b)

Class Time	Had Proper Sleep	Weather	Attended ML Class
Morning	YES	COOL	YES
Morning	NO	RAINY	NO
Morning	NO	COOL	YES
Morning	YES	HOT	YES
Noon	YES	COOL	YES
Noon	NO	HOT	NO
Noon	NO	COOL	NO
Noon	YES	HOT	YES
Afternoon	YES	COOL	YES
Afternoon	NO	RAINY	NO
Afternoon	NO	HOT	NO
Afternoon	YES	HOT	YES

Table 1: Given Dataset

Step 1: Calculating the Entropy of the Entire Dataset

- Total 'Yes' for attending the ML class: 7
- Total 'No' for attending the ML class: 5
- Total instances: 12

$$\text{Entropy}(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

The entropy of the entire dataset is:

$$\text{Entropy}(S) = -\frac{7}{12}\log_2\left(\frac{7}{12}\right) - \frac{5}{12}\log_2\left(\frac{5}{12}\right) = 0.98$$

Step 2: Information Gain of Each Feature

1. Class Time

Feature **Class Time** has three attributes: Morning, Noon, and Afternoon

- Entropy for Morning:

$$\text{Entropy}(S_{\text{Morning}}) = -\frac{3}{4}\log_2\left(\frac{3}{4}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) = 0.811$$

- Entropy for Noon:

$$\text{Entropy}(S_{\text{Noon}}) = -\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right) = 1$$

- Entropy for Afternoon:

$$\text{Entropy}(S_{\text{Afternoon}}) = -\frac{2}{4}\log_2\left(\frac{2}{4}\right) - \frac{2}{4}\log_2\left(\frac{2}{4}\right) = 1$$

Now, the Information Gain for Class Time is:

$$\text{IG(Class Time)} = 0.957 - \frac{4}{12}(0.811) - \frac{4}{12}(1) - \frac{4}{12}(1) = \mathbf{0.04}$$

2. Proper Sleep

Feature **Proper Sleep** has two attributes: Yes and No

- Entropy for Proper Sleep = Yes:

$$\text{Entropy}(S_{\text{Yes}}) = -\frac{6}{6}\log_2\left(\frac{6}{6}\right) = 0$$

- Entropy for Proper Sleep = No:

$$\text{Entropy}(S_{\text{No}}) = -\frac{1}{6}\log_2\left(\frac{1}{6}\right) - \frac{5}{6}\log_2\left(\frac{5}{6}\right) = 0.65$$

Now, the Information Gain for Proper Sleep is:

$$\text{IG(Proper Sleep)} = 0.957 - \frac{6}{12}(0) - \frac{6}{12}(0.65) = \mathbf{0.655}$$

3. **Weather**
Feature **Weather** has three attributes: Hot, Cool, and Rainy

- Entropy for Hot:

$$\text{Entropy}(S_{\text{Hot}}) = -\frac{3}{5}\log_2\left(\frac{3}{5}\right) - \frac{2}{5}\log_2\left(\frac{2}{5}\right) = 0.97$$

- Entropy for Cool:

$$\text{Entropy}(S_{\text{Cool}}) = -\frac{4}{5}\log_2\left(\frac{4}{5}\right) - \frac{1}{5}\log_2\left(\frac{1}{5}\right) = 0.72$$

- Entropy for Rainy:

$$\text{Entropy}(S_{\text{Rainy}}) = 0$$

Now, the Information Gain for Weather is:

$$\text{IG(Weather)} = 0.957 - \frac{5}{12}(0.97) - \frac{5}{12}(0.72) - \frac{2}{12}(0) = \mathbf{0.27}$$

Step 3: Choosing the Root Node

Among all attributes, the attribute Proper Sleep has the **highest** Information Gain (**0.632**). Hence, Proper Sleep is selected as the **root node** of the decision tree.

Step 4: Building Rest of the Tree

When Proper Sleep is "Yes", the entropy of the subset is 0, so all outcomes are "Yes" (Attended ML Class). When Proper Sleep is "No", further splitting is done based on Class Time or Weather.

Class Time	Had Proper Sleep	Weather	Attended ML Class
Morning	NO	RAINY	NO
Morning	NO	COOL	YES
Noon	NO	HOT	NO
Noon	NO	COOL	NO
Afternoon	NO	RAINY	NO
Afternoon	NO	HOT	NO

Table 2: Subset of Data based on "had proper sleep" feature

Now, Yes = 1, No = 5
The Entropy of Table 2 is:

$$\text{Entropy}(S_{No}) = -\frac{1}{6}\log_2\left(\frac{1}{6}\right) - \frac{5}{6}\log_2\left(\frac{5}{6}\right)$$

$$\text{Entropy}(S_{No}) = -0.1667\log_2(0.1667) - 0.8333\log_2(0.8333) \approx 0.65$$

4.1 Information Gain for Class Time (when Proper Sleep = No)

Feature **Class Time** has three attributes: Morning, Noon, and Afternoon

- Entropy for Morning:

$$\text{Entropy}(S_{Morning}) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$$

- Entropy for Noon:

$$\text{Entropy}(S_{Noon}) = 0 \quad \text{(since all outcomes are "No")}$$

- Entropy for Afternoon:

$$\text{Entropy}(S_{Afternoon}) = 0 \quad \text{(since all outcomes are "No")}$$

Now, the Information Gain for Class Time is:

$$IG(Class\ Time) = 0.65 - \frac{2}{6}(1) + \frac{2}{6}(0) + \frac{2}{6}(0) = 0.333 = \mathbf{0.317}$$

4.2 Information Gain for Weather (when Proper Sleep = No)

Feature **Weather** has three attributes: Hot, Cool, and Rainy

- Entropy for Hot:

$$\text{Entropy}(S_{Hot}) = 0 \quad \text{(since all outcomes are "No")}$$

- Entropy for Cool:

$$\text{Entropy}(S_{Cool}) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{2}\log_2\left(\frac{1}{2}\right) = 1$$

- Entropy for Rainy:

$$\text{Entropy}(S_{Rainy}) = 0 \quad \text{(since all outcomes are "No")}$$

Now, the Information Gain for Weather is:

$$IG(Weather) = 0.65 - \frac{2}{6}(0) + \frac{2}{6}(1) + \frac{2}{6}(0) = 0.333 = \mathbf{0.317}$$

Information Gained for both Class Time and Weather is the same; hence, either can be chosen.

Solution (d)

(a) Probability Estimates

For Spam (class = 1):

$$P(\text{buy} = 0|\text{Spam}) = \frac{0}{2} = 0$$

$$P(\text{buy} = 1|\text{Spam}) = \frac{2}{2} = 1$$

$$P(\text{cheap} = 0|\text{Spam}) = \frac{1}{2} = 0.5$$

$$P(\text{cheap} = 1|\text{Spam}) = \frac{1}{2} = 0.5$$

For Non-Spam (class = 0):

$$P(\text{buy} = 0|\text{Non-Spam}) = \frac{1}{2} = 0.5$$

$$P(\text{buy} = 1|\text{Non-Spam}) = \frac{1}{2} = 0.5$$

$$P(\text{cheap} = 0|\text{Non-Spam}) = \frac{1}{2} = 0.5$$

$$P(\text{cheap} = 1|\text{Non-Spam}) = \frac{1}{2} = 0.5$$

(b) Posterior Probabilities

Using the Naive Bayes formula:

$$P(\text{Spam}|\text{buy} = 0, \text{cheap} = 1) = P(\text{Spam}) \times P(\text{buy} = 0|\text{Spam}) \times P(\text{cheap} = 1|\text{Spam})$$

$$P(\text{Non-Spam}|\text{buy} = 0, \text{cheap} = 1) = P(\text{Non-Spam}) \times P(\text{buy} = 0|\text{Non-Spam}) \times P(\text{cheap} = 1|\text{Non-Spam})$$

For Spam:

$$P(\text{Spam}|\text{buy} = 0, \text{cheap} = 1) \propto \frac{2}{4} \times 0 \times \frac{1}{2} = 0 \times 0.5 = 0$$

For Non-Spam:

$$P(\text{Non-Spam}|\text{buy} = 0, \text{cheap} = 1) \propto \frac{2}{4} \times \frac{1}{2} \times \frac{1}{2} = \frac{0.5 \times 0.25}{0.125} = 1$$

(c) Problem with Zero Probabilities and Solution

The main problem with the Zero Probability of P(buy = 0 | Spam) = 0 is it leads to the entire posterior probability for Spam being 0. The issue of zero probabilities can be resolved using **Laplace Smoothing**. This technique adds 1 to all frequency counts:

$$P(\text{buy} = 0|\text{Spam}) = \frac{(\text{count of Feature 1 in spam} = 0) + 1}{\text{total spam emails} + \text{possible outcome of Feature 1}}$$

Therefore,

$$P(\text{buy} = 0|\text{Spam}) = \frac{0 + 1}{2 + 2} = \frac{1}{4} = 0.25$$

2 Section C (Algorithm implementation using packages)

Part A: EDA

a) Overview of the Dataset

Label	Count
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

Table 3: Class Distribution

Metric	Width	Height
count	12600.000000	12600.000000
mean	260.381032	196.573571
std	39.919281	35.281402
min	84.000000	84.000000
25%	254.000000	181.000000
50%	275.000000	183.000000
75%	276.000000	194.000000
max	478.000000	318.000000

Table 4: Image Size Statistics

From the dataset, it is clearly visible that all classes are perfectly balanced, with every class having an equal number of images. The average size of images in the dataset is 260 x 196. However, some images are very small, and some are very large, so we need to filter out those images from the dataset to improve model performance.

b) Visual Representation of Distribution of Image Sizes

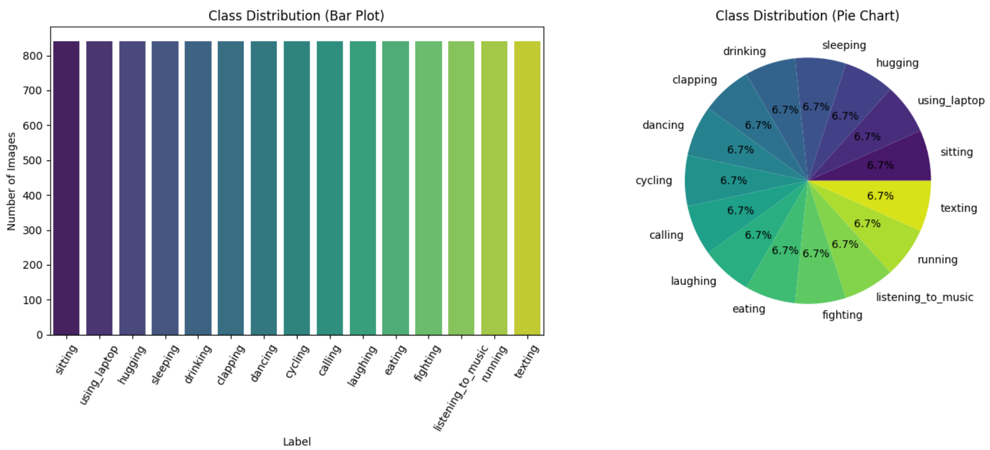

Figure 1: Class Distribution of Images

The class distribution of images is equal. But by displaying some random images of each class, I observed that some images have watermarks and text, which might affect the overall accuracy of our model

c) Check Class Imbalance

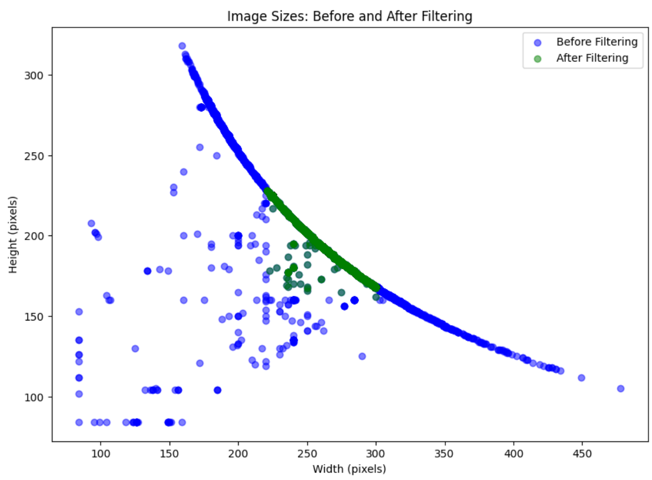

Figure 2: Comparison of Image Sizes Before and After Filtering

In the initial dataset, there is no class imbalance, but after data filtration, there might be chances of class imbalance, so we have to take care of that before feeding data to the model.

Part B: Feature Extraction

I extracted features using four different types of techniques:-

1. **Histogram of Oriented Gradients (HOG)**: For extracting texture features.
2. **Color Histograms**:- To distinguish objects in the image based on colour composition.
3. **Scale Invariant Feature Transform (SIFT)**:- To match objects in images with different viewpoints.
4. **Canny Edge Detection**:- To extract object boundaries.

By the combination of the above four techniques, I extracted **100656** different features. After that, I split the features into training and testing sets; then, on the testing set, I first applied standardization, then PCA for Dimensionality Reduction. After that, I applied Features resampling using SMOTE, and then our features were ready to be passed in the model for training.

Part C: Model Selection and Evaluation

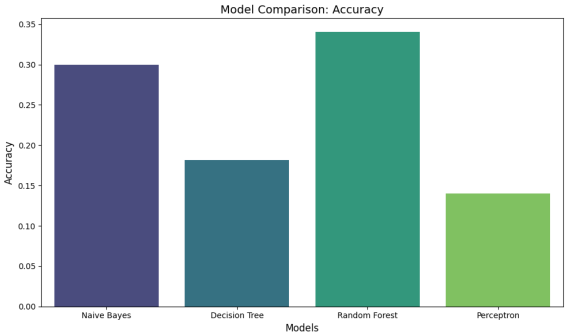

Figure 3: Models vs Accuracy Graph

Model	Accuracy
Naive Bayes	0.299376
Decision Tree	0.181913
Random Forest	0.351937
Perceptron	0.139813

Table 5: Models Accuracy Table

From the Results, I observed that Random Forest performed the best among the models due to its ensemble nature. Second is Naive Bayes, which might struggle due to highly correlated data. The decision tree and Perceptron model perform very poorly because the Decision tree will get overfitted, and the perceptron model is very simple in nature, so it can't handle the complex image data.