دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده برق و کامپیوتر

یادگیری ماشین

تمرین شماره 1

نام و نام خانوادگی
سید محمد جزایری

شماره دانشجویی
810101561

آبان 1404

فهرست

فهرست شکل‌ها 2
چکیده 3
پرسش 1 - عنوان پرسش.. 4
1-1. عنوان بخش.. 4
2-1. عنوان بخش 4
3-1. عنوان بخش 4

پرسش 2 - عنوان پرسش......5
.1-2 عنوان بخش......5
.2-2 عنوان بخش......5
پرسش 3 - عنوان پرسش......6
.1-3 عنوان بخش......6
.2-3 عنوان بخش......6
پرسش 4 - عنوان پرسش......7
.1-4 عنوان بخش......7
.2-4 عنوان بخش......7
.3-4 عنوان بخش......7
پرسش 5 - عنوان پرسش......7
.1-5 عنوان بخش......7
.2-5 عنوان بخش......8
پرسش 6 - عنوان پرسش......8
پرسش 7 - عنوان پرسش......8
.1-7 عنوان بخش......8
.2-7 عنوان بخش......9
پرسش 8 - عنوان پرسش......10
.2-8 عنوان بخش......10
.3-8 عنوان بخش......11
.4-8 عنوان بخش......11
.5-8 عنوان بخش......11
.6-8 عنوان بخش......12
پرسش 9 - عنوان پرسش......12
.1-9 عنوان بخش......12
.2-9 عنوان بخش......12
.3-9 عنوان بخش......13
.4-9 عنوان بخش......13
.5-9 عنوان بخش......14

فهرست شکل‌ها

شکل 1.The distribution of data and the discriminant function 6
شکل 2.The result of the calculations. 9
شکل 3. 9
شکل 4. 10
شکل 5. 10
شکل 6.Probability of ham and spam 11
شکل 7.Top ham and spam words 11
شکل 8.Accuracy on test and confusion matrix 11
شکل 9.Accuracy Vs. m 12

شکل 10.Results of the blue ratio classifier 13
شکل 11.Results of the white ratio classifier.. 13
شکل 12.Part 3 13
شکل 13.Part 1.. 14
شکل 14.s33 14
شکل 15.c1 misclassified by white 14
شکل 16.c36 misclassified by blue 15

چکیده

این تمرین در مورد تصمیم گیری با مبنای بیز است. در این تمرین با انواع مختلف آن(ریسک کمینه و خطای کمینه) کار می کنیم و همچنین از روش maximum likelihood برای تخمین پارامترها استفاده می کنیم. در تمرین های عملی هم از Naive Bayes استفاده می کنیم و قدرت خوبش با توجه به فرض بسیار ساده‌اش را در کار با متن می بینیم.

پرسش 1 - عنوان پرسش

.1-1 عنوان بخش

the risk is defined as $R_{Rand}(x) = \sum_{k=1}^{K} p(\omega_k | x) * \sum_{j\neq k}^{K} p(\omega_j | x)$

The first summation is the probability of choosing class k and the second summation is the loss incurred by choosing this class, since the choice is random and the loss is zero-one the loss is also a random variable. and note

$$\sum_{j\neq k}^{K} p(\omega_j | x) = 1 - p(\omega_k | x)$$

so we simplify to:

$$\sum_{k=1}^{K} p(\omega_k | x) * (1 - p(\omega_k | x)) = \sum_{k=1}^{K} p(\omega_k | x) - p(\omega_k | x)^2 = 1 - \sum_{k=1}^{K} p(\omega_k | x)^2$$

whereas the Bayesian risk is

$1 - max_i\, p(\omega_i | x)$.

.2-1 عنوان بخش

now to prove the risk of the Bayesian classifier is less we have to prove:

$max_i\, p(\omega_i | x) \leq \sum_{k=1}^{K} p(\omega_k | x)^2$ assume $m = max_i\, p(\omega_i | x)$

we know

$m \geq p(\omega_k | x) \ \ \forall k$

now multiply both sides by $p(\omega_k | x)$ and we get

$m * p(\omega_k | x) \ \geq p(\omega_k | x)^2$

now we sum over k

$$\sum_{k=1}^{K} m * p(\omega_k | x) \geq \sum_{k=1}^{K} p(\omega_k | x)^2$$

and since m is a constant we take it out of the sum and we get

$$\sum_{k=1}^{K} p(\omega_k | x)^2 \leq m$$

which proves $R_{Rand} \geq R_{Bayesian}$

3-1. عنوان بخش

these two classifiers will be the same if and only if

$$max_i\, p(\omega_i \mid x) = \sum_{k=1}^{K} p(\omega_k \mid x)^2$$

from our inequality chain we have

$$m * p(\omega_k \mid x) = p(\omega_k \mid x)^2 \ \forall\, k$$

which happens when either $p(\omega_k \mid x) = 0$ or $p(\omega_k \mid x) = m$

the second way implies that we have a uniform distribution with each class having the probability of $\frac{1}{K}$

پرسش 2 - عنوان پرسش

1-2. عنوان بخش

We use the likelihood ratio.

$$\frac{p(x \mid \omega_1)}{p(x \mid \omega_2)} > \frac{\lambda_{12} p(\omega_2)}{\lambda_{21} p(\omega_1)}$$

we apply ln to both sides. First we calculate the left-hand side.

$$ln \frac{p(x \mid \omega_1)}{p(x \mid \omega_2)} = - \frac{x^2 + (x-1)^2}{2\sigma^2} = \frac{1-2x}{2\sigma^2}$$

so the decision is ω_1 iff

$$\frac{1-2x}{2\sigma^2} > ln \frac{\lambda_{12} p(\omega_2)}{\lambda_{21} p(\omega_1)} \quad \text{we solve for x}$$

$$x < \frac{1}{2} - \sigma^2 ln \frac{\lambda_{12} p(\omega_2)}{\lambda_{21} p(\omega_1)}$$

and to minimize the threshold

$$\tau = \frac{1}{2} - \sigma^2 ln \frac{\lambda_{12} p(\omega_2)}{\lambda_{21} p(\omega_1)}$$

2-2. عنوان بخش

a) since the covariance matrices are equal for both distributions we know from the class notes that the discriminant functions are minimum distant classifiers, hence

$$g_i = -\frac{1}{2}(x - \mu_i)^T \ \Sigma^{-1} (x - \mu_i) + ln(p(\omega_i))$$

with $\Sigma = 3I$ we get:

$$g_i = -\frac{1}{6}\left|\left|x - \mu_i\right|\right|^2 + ln(p(\omega_i))$$

b) we set $g_1 = g_2$ which yields:

$$-\frac{1}{6}\left|\left|x - \mu_1\right|\right|^2 + ln(p(\omega_1)) = -\frac{1}{6}\left|\left|x - \mu_2\right|\right|^2 + ln(p(\omega_2))$$

$$\frac{1}{6}(\left\|x - \mu_2\right\|^2 - \left\|x - \mu_1\right\|^2) = ln(\frac{p(\omega_2)}{p(\omega_1)})$$

after some calculations we get:

$16x_2 - 12x_1 = 6ln(0.66) + 28$

c) if we change the priors we'll move the line and create another line parallel to the one we just calculated— if we change the covariance matrix but let both classes have the same matrix our boundary won't change but if we assign each one a different matrix then we'll have a quadratic form and our boundary will no longer be linear.

d) As we can see in Figure 1. The line separates our two distributions pretty well.

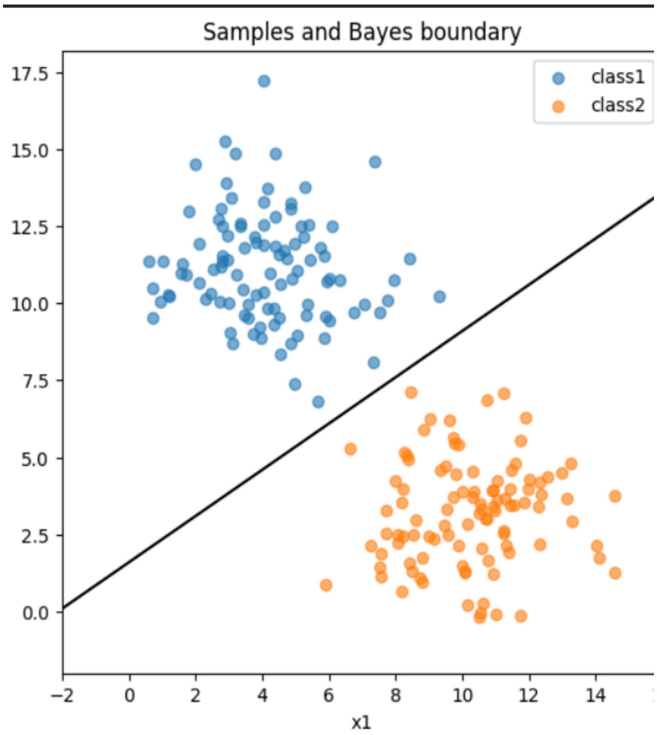


Figure 1. The distribution of data and the discriminant function

پرسش 3 - عنوان پرسش

3-1. عنوان بخش

Each x_j and y are Bernoulli random variables, the only parameter in such random variables is the probability of success (φ in this case).

$$l(\varphi) = \sum_{i=1}^{N} logP(y^{(i)}; \varphi_i) + \sum_{j=1}^{d}\sum_{i=1}^{N} logP(x_j^{(i)}|y^{(i)}; \varphi_{j|y})$$

after plugging in the values we get:

$$l(\varphi) = \sum_{i=1}^{N} log(\varphi_y^{y^{(i)}} * (1 - \varphi_y)^{1-y^{(i)}}) + \sum_{j=1}^{d}\sum_{i=1}^{N} log(\varphi_{j|y}^{x_j^{(i)}} * (1 - \varphi_{j|y})^{1-x_j^{(i)}})$$

$$l(\varphi) = \sum_{i=1}^{N} y^{(i)} log\varphi_i + \sum_{i=1}^{N} (1 - y^{(i)}) log(1 - \varphi_i) + \sum_{j=1}^{d}\sum_{i=1}^{N} x_j^{(i)} log\varphi_{j|y} +$$

$$\sum_{j=1}^{d}\sum_{i=1}^{N}(1 - x_j^{(i)})\ log(1 - \varphi_{j|y})$$

2-3. عنوان بخش

to compute the optimal values for our parameters we differentiate and set to zero.

$$\frac{\partial l(\varphi)}{\partial \varphi_y} = \sum_{i=1}^{N}\left(\frac{y^{(i)}}{\varphi_y} - \frac{1-y^{(i)}}{1-\varphi_y}\right) = 0$$

$$\widehat{\varphi_y} = \frac{1}{N}\sum_{i=1}^{N} y^{(i)}$$

This means count how many instances are 1 and divide by the number of all the samples(empirical frequencies).

$$\frac{\partial l(\varphi)}{\partial \varphi_{j|y=y_0}} = \sum_{j=1}^{d}\left[\sum_{i:y^{(i)}=y_0}\frac{x_j^{(i)}}{\varphi_{j|y=y_0}} - \frac{1-x_j^{(i)}}{1-\varphi_{j|y=y_0}}\right] = 0$$

$$\widehat{\varphi_{j|y=y_0}} = \frac{\sum_{i=1}^{N} 1\{y^{(i)}=y_0\}x_j^{(i)}}{\sum_{i=1}^{N} 1\{y^{(i)}=y_0\}}$$

This means count how many training examples with label b have feature j=1 and divide by the total number of examples with label b(empirical frequencies).

پرسش 4 - عنوان پرسش

1-4. عنوان بخش

$l(\lambda) = \prod_{i=1}^{N} e^{-\lambda}\frac{\lambda^{x_i}}{x_i!}$ As the question hinted, we apply log to both sides.

$$l(\lambda) = \sum_{i=1}^{N} -\lambda + x_i log(\lambda) - const.$$

$$\frac{\partial l(\lambda)}{\partial \lambda} = -N + \frac{1}{\lambda}\sum_{i=1}^{N} x_i = 0$$

$$\lambda^* = \frac{1}{N}\sum_{i=1}^{N} x_i$$

2-4. عنوان بخش

$$\lambda^* = \frac{13}{5} = 2.6$$

$P(X = 6 \mid \lambda^* = 2.6) = e^{-2.6}\frac{(-2.6)^6}{6!} \simeq 0.03187$

3-4. عنوان بخش

Given this new info our new λ^* is 3.5

$P(X = 6 \mid \lambda^* = 3.5) = e^{-3.5}\frac{(-3.5)^6}{6!} \simeq 0.0771$

پرسش 5 - عنوان پرسش

1-5. عنوان بخش

First of all we rewrite the formula in the question.
$\epsilon^{(i)} = y^{(i)} - \theta^{T} x^{(i)}$
Now we can obtain the probability required.
$P(y^{(i)} | x^{(i)}; \Theta, \sigma^{2}) = \frac{1}{\sqrt{2\pi}\,\sigma} exp\{- \frac{1}{2\sigma^{2}} (y^{(i)} - \theta^{T} x^{(i)})^{2} \}$
hence the likelihood is:
$l(\Theta, \sigma^{2}) = \frac{1}{\sqrt{2\pi}\,\sigma} \prod_{i=1}^{N} exp\{- \frac{1}{2\sigma^{2}} (y^{(i)} - \theta^{T} x^{(i)})^{2} \}$
log-likelihood is:
$l(\Theta, \sigma^{2}) = log(\frac{1}{\sqrt{2\pi}\,\sigma}) - \frac{1}{\sigma^{2}} \sum_{i=1}^{N} ((y^{(i)} - \theta^{T} x^{(i)})$

2-5. عنوان بخش

It is fairly obvious that $log(\frac{1}{\sqrt{2\pi}\,\sigma})$ is a constant with respect to σ so we'll call it A. Now in order to maximize $l(\Theta, \sigma^{2})$ we must minimize $\frac{1}{\sigma^{2}} \sum_{i=1}^{N} ((y^{(i)} - \theta^{T} x^{(i)})$ because it is being subtracted from the constant A, And this term is the Mean Squared Error.

پرسش 6 - عنوان پرسش

$P(\omega_{1}|x) > P(\omega_{2}|x) \Rightarrow \frac{P(x|\omega_{1})P(\omega_{1})}{P(x)} > \frac{P(x|\omega_{2})P(\omega_{2})}{P(x)} \Rightarrow \frac{P(x|\omega_{1})}{P(x|\omega_{2})} > \frac{P(\omega_{2})}{P(\omega_{1})}$

$\frac{\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^{2}}(x-\mu_{1})^{2}}}{\frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{1}{2\sigma^{2}}(x-\mu_{2})^{2}}} > \frac{P(\omega_{2})}{P(\omega_{1})}$ Apply log.

$\frac{-1}{2\sigma^{2}} \left[(x - \mu_{1})^{2} - (x - \mu_{2})^{2} \right] > log \frac{P(\omega_{2})}{P(\omega_{1})}$ Simplify for optimal x.

$x(\mu_{1} - \mu_{2}) + \frac{1}{2}(\mu_{2}^{2} - \mu_{1}^{2}) + \sigma^{2} log \frac{P(\omega_{1})}{P(\omega_{2})} = 0$

$x^{*} = \frac{1}{2}(\mu_{1} + \mu_{2}) + \frac{\sigma^{2}}{\mu_{1} - \mu_{2}} log(\frac{P(\omega_{2})}{P(\omega_{1})})$ if the priors were equal the second term would've equaled zero and the optimal point would've been the middle point of the connecting line of the two means.
If we increase $P(\omega_{1})$ then the border will go farther from the mean of ω_{1} which means its decision space will grow which means choosing the class ω_{1} will be more probable.

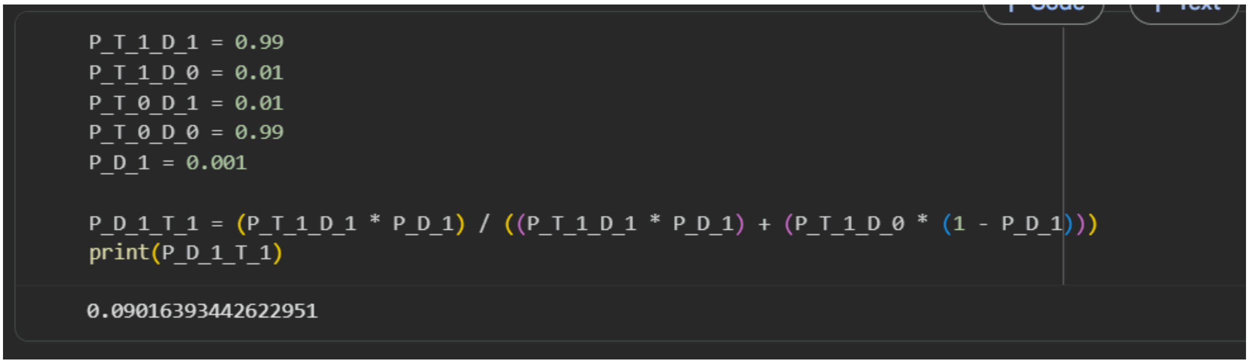
پرسش 7 - عنوان پرسش

1-7. عنوان بخش

We will use Bayes' theorem for this part

$$P(D = 1 \mid T = 1) = \frac{P(T=1 \mid D=1)\, P(D=1)}{P(T=1)}$$

The details are explained in the notebook.



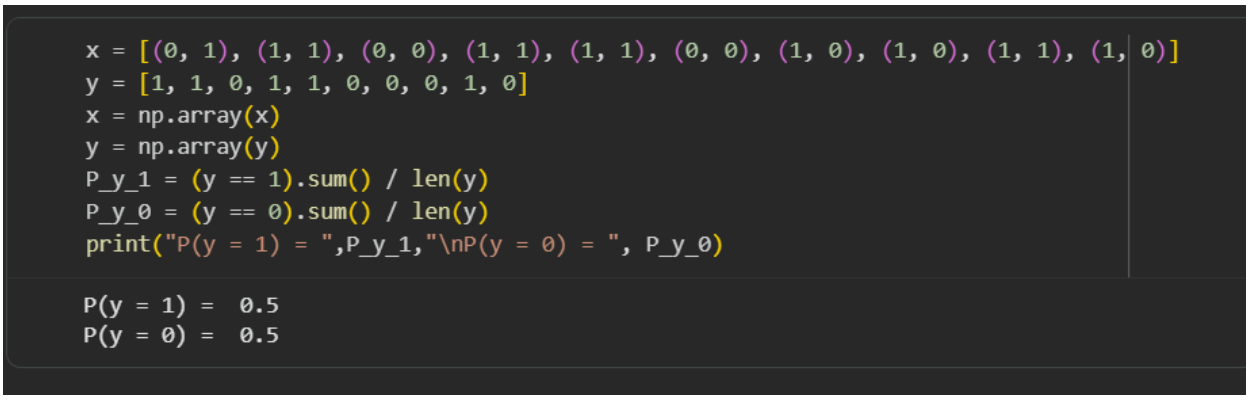
```
P_T_1_D_1 = 0.99
P_T_1_D_0 = 0.01
P_T_0_D_1 = 0.01
P_T_0_D_0 = 0.99
P_D_1 = 0.001

P_D_1_T_1 = (P_T_1_D_1 * P_D_1) / ((P_T_1_D_1 * P_D_1) + (P_T_1_D_0 * (1 - P_D_1)))
print(P_D_1_T_1)
```

```
0.09016393442622951
```

Figure 2. The result of the calculations.

2-7. عنوان بخش

a) We have to count the number of times y = a (a = 1 or 0) has occurred and divide by the number of all instances (10).

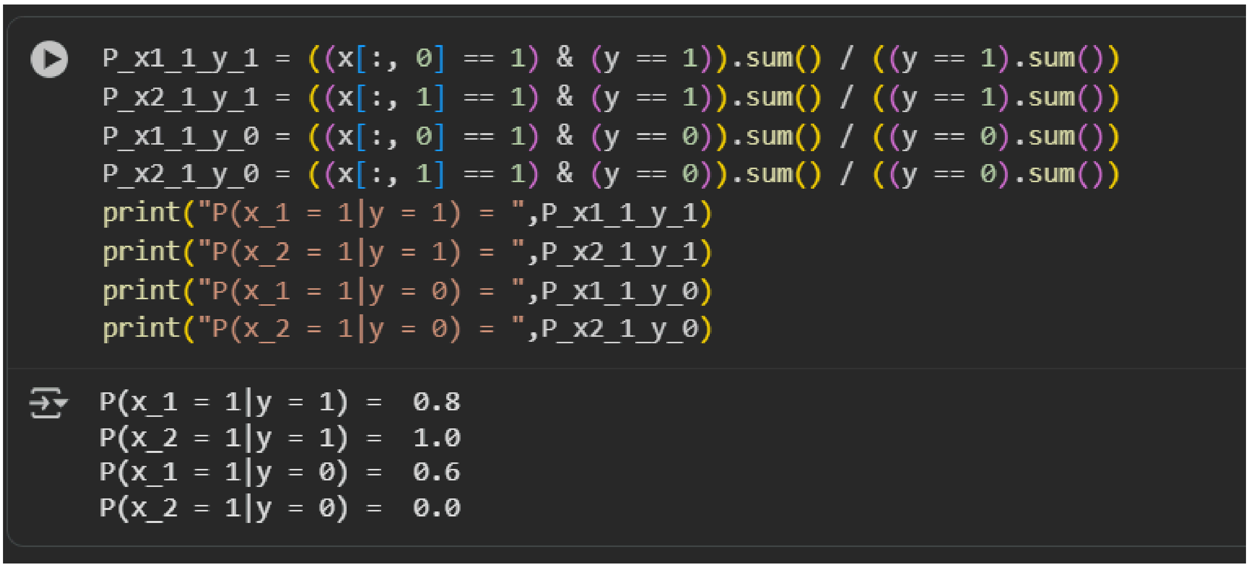


```
x = [(0, 1), (1, 1), (0, 0), (1, 1), (1, 1), (0, 0), (1, 0), (1, 0), (1, 1), (1, 0)]
y = [1, 1, 0, 1, 1, 0, 0, 0, 1, 0]
x = np.array(x)
y = np.array(y)
P_y_1 = (y == 1).sum() / len(y)
P_y_0 = (y == 0).sum() / len(y)
print("P(y = 1) = ",P_y_1,"\nP(y = 0) = ", P_y_0)
```

```
P(y = 1) =  0.5
P(y = 0) =  0.5
```

Figure 3.

b) For each probability we count the number of times x = a when y = b in the instances.



```
P_x1_1_y_1 = ((x[:, 0] == 1) & (y == 1)).sum() / ((y == 1).sum())
P_x2_1_y_1 = ((x[:, 1] == 1) & (y == 1)).sum() / ((y == 1).sum())
P_x1_1_y_0 = ((x[:, 0] == 1) & (y == 0)).sum() / ((y == 0).sum())
P_x2_1_y_0 = ((x[:, 1] == 1) & (y == 0)).sum() / ((y == 0).sum())
print("P(x_1 = 1|y = 1) = ",P_x1_1_y_1)
print("P(x_2 = 1|y = 1) = ",P_x2_1_y_1)
print("P(x_1 = 1|y = 0) = ",P_x1_1_y_0)
print("P(x_2 = 1|y = 0) = ",P_x2_1_y_0)
```

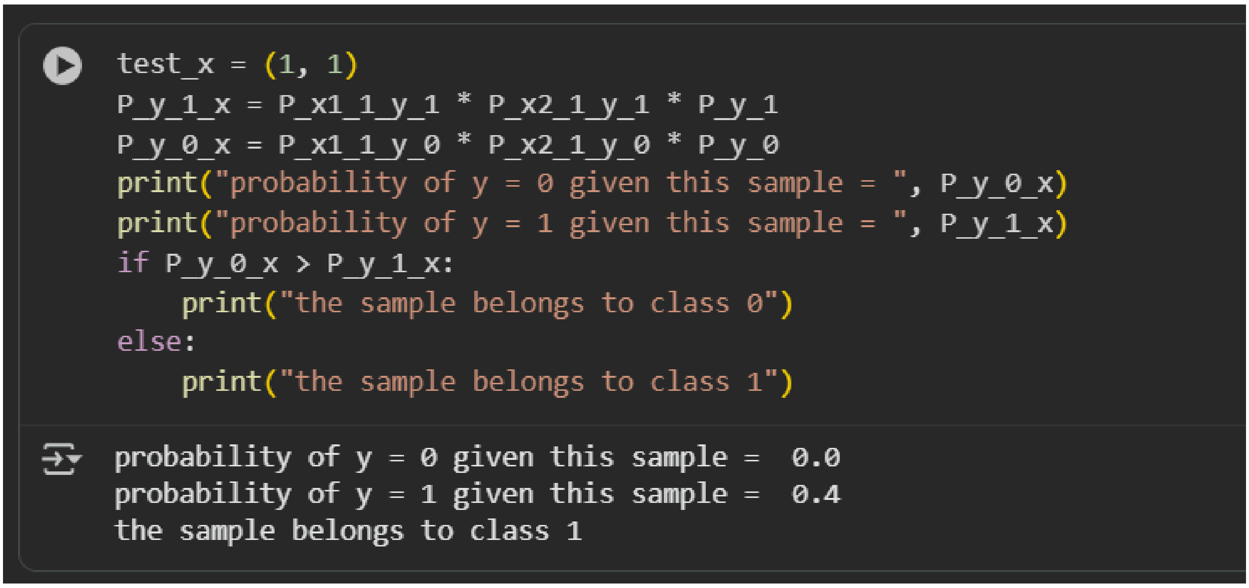
```
P(x_1 = 1|y = 1) =  0.8
P(x_2 = 1|y = 1) =  1.0
P(x_1 = 1|y = 0) =  0.6
P(x_2 = 1|y = 0) =  0.0
```

Figure 4.

c) Naive Bayes' assumption:

$$P(x_1, ..., x_n \,|\, Label) \;=\; P(Label) \prod_{i=1}^{n} P(x_i \,|\, Label)$$

We assign the sample to the class with the highest probability.



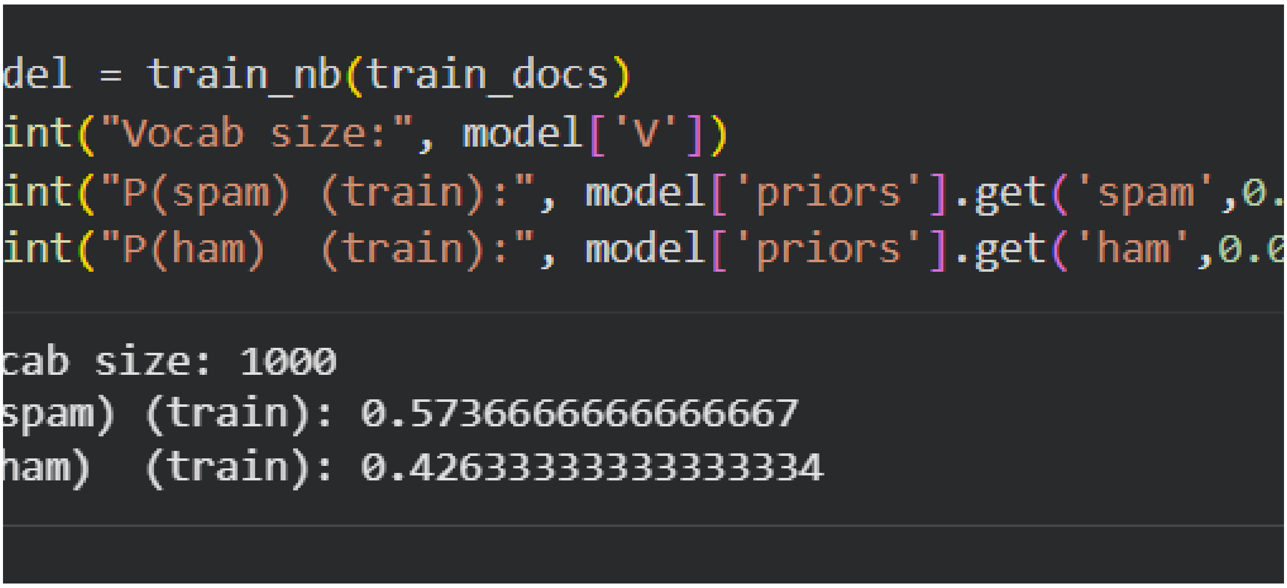
```
test_x = (1, 1)
P_y_1_x = P_x1_1_y_1 * P_x2_1_y_1 * P_y_1
P_y_0_x = P_x1_1_y_0 * P_x2_1_y_0 * P_y_0
print("probability of y = 0 given this sample = ", P_y_0_x)
print("probability of y = 1 given this sample = ", P_y_1_x)
if P_y_0_x > P_y_1_x:
    print("the sample belongs to class 0")
else:
    print("the sample belongs to class 1")
```

```
probability of y = 0 given this sample =  0.0
probability of y = 1 given this sample =  0.4
the sample belongs to class 1
```

Figure 5.

پرسش 8 - عنوان پرسش

2-8. عنوان بخش

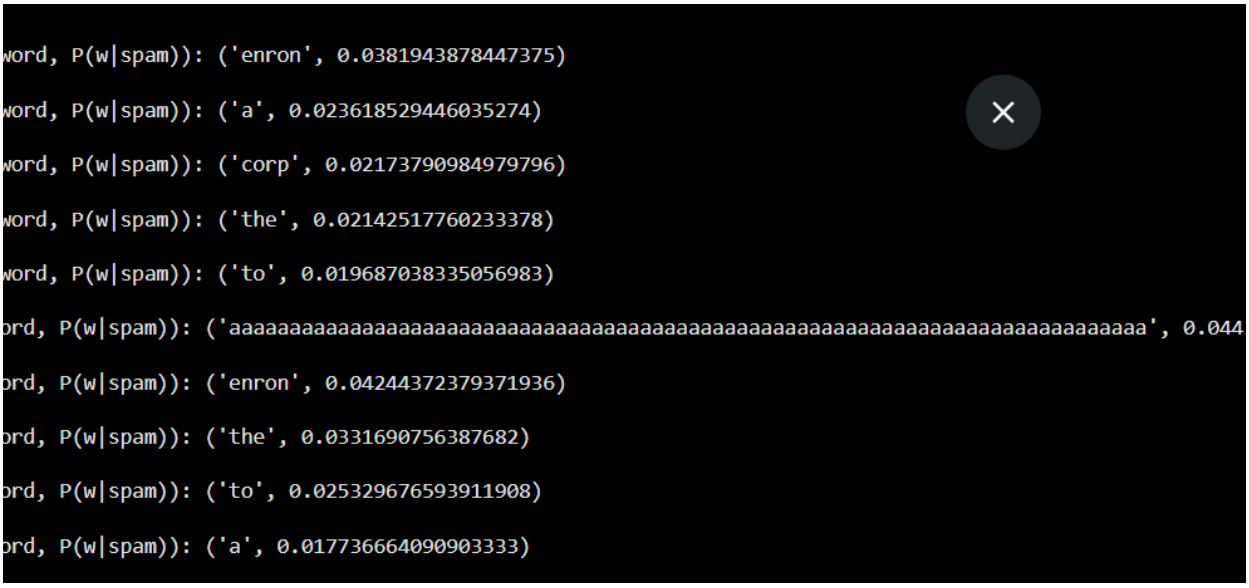


```
del = train_nb(train_docs)
int("Vocab size:", model['V'])
int("P(spam) (train):", model['priors'].get('spam',0.
int("P(ham)  (train):", model['priors'].get('ham',0.0
```

```
cab size: 1000
spam) (train): 0.5736666666666667
ham)  (train): 0.4263333333333334
```

Figure 6. Probability of ham and spam

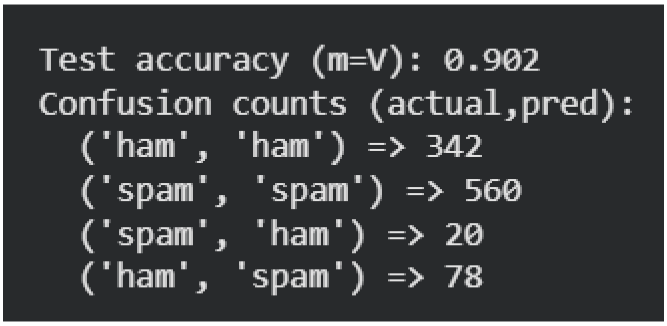
8-3. عنوان بخش



```
word, P(w|spam)): ('enron', 0.0381943878447375)
word, P(w|spam)): ('a', 0.023618529446035274)
word, P(w|spam)): ('corp', 0.02173790984979796)
word, P(w|spam)): ('the', 0.02142517760233378)
word, P(w|spam)): ('to', 0.019687038335056983)
ord, P(w|spam)): ('aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa', 0.044
ord, P(w|spam)): ('enron', 0.04244372379371936)
ord, P(w|spam)): ('the', 0.0331690756387682)
ord, P(w|spam)): ('to', 0.025329676593911908)
ord, P(w|spam)): ('a', 0.017736664090903333)
```

Figure 7. Top ham and spam words

8-4. عنوان بخش



```
Test accuracy (m=V): 0.902
Confusion counts (actual,pred):
  ('ham', 'ham') => 342
  ('spam', 'spam') => 560
  ('spam', 'ham') => 20
  ('ham', 'spam') => 78
```

Figure 8. Accuracy on test and confusion matrix

8-5. عنوان بخش

This is our formula:

$$P(\omega_i \mid C) = \frac{n_i + m * p}{n + m} \qquad p = \frac{1}{|V|}$$

Basically when we make m larger we are omitting n_i and n and we are mostly relying on the priors.

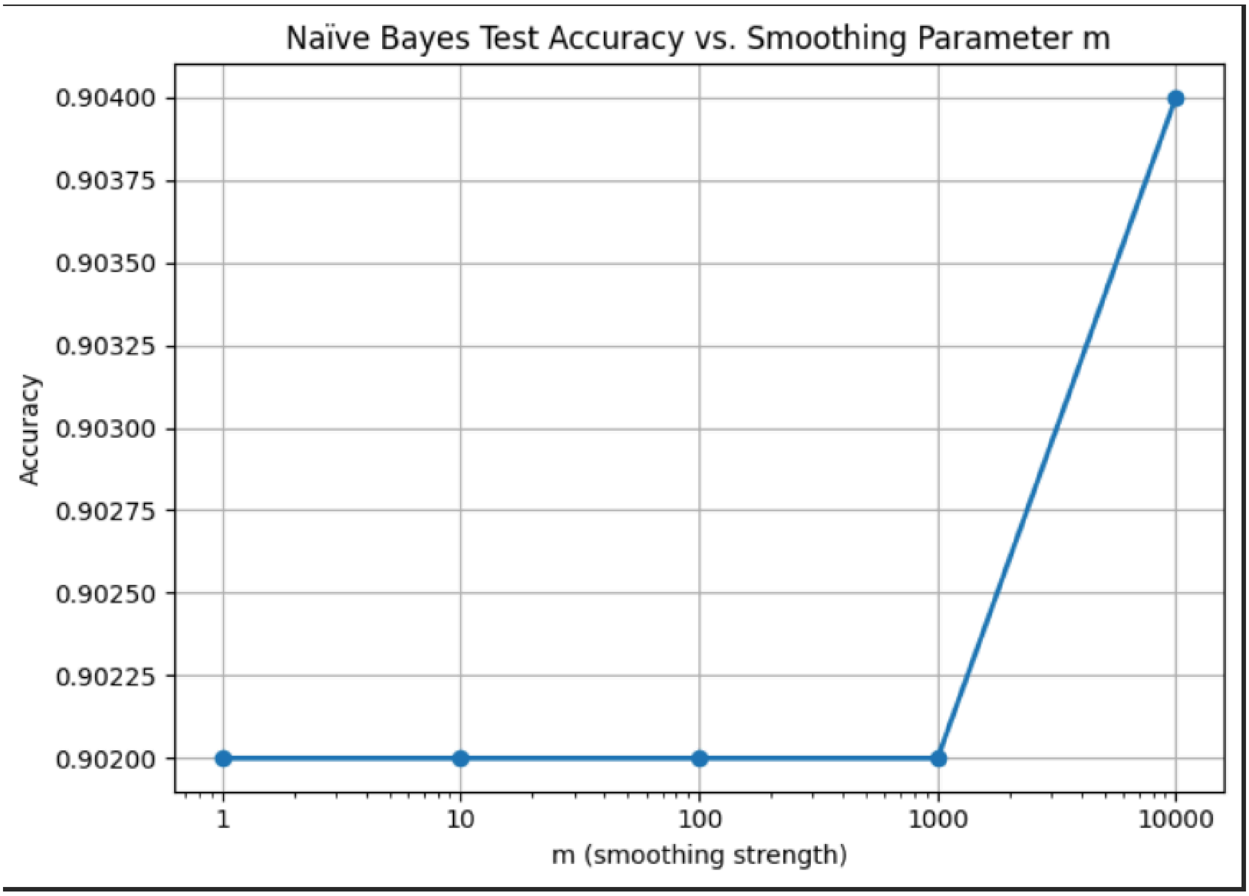


Figure 9. Accuracy Vs. m

6-8. عنوان بخش

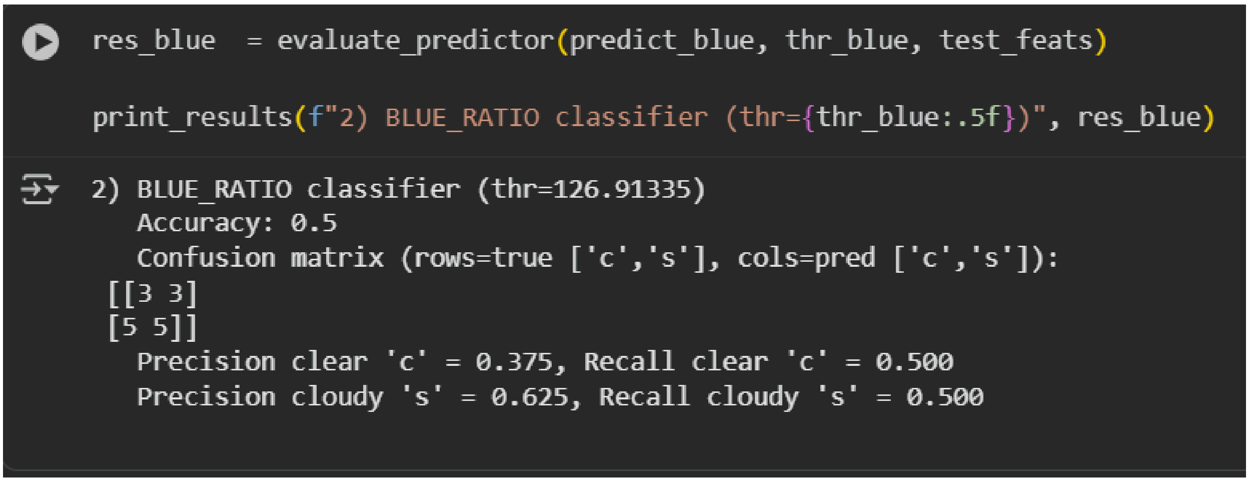
I’d use more words with high ham probability and avoid ones with high spam probability.

پرسش 9 - عنوان پرسش

1-9. عنوان بخش

Since the sky is blue I will find the ratio of the blue pixels in clear images and use the mean as a threshold on the test data.

2-9. عنوان بخش



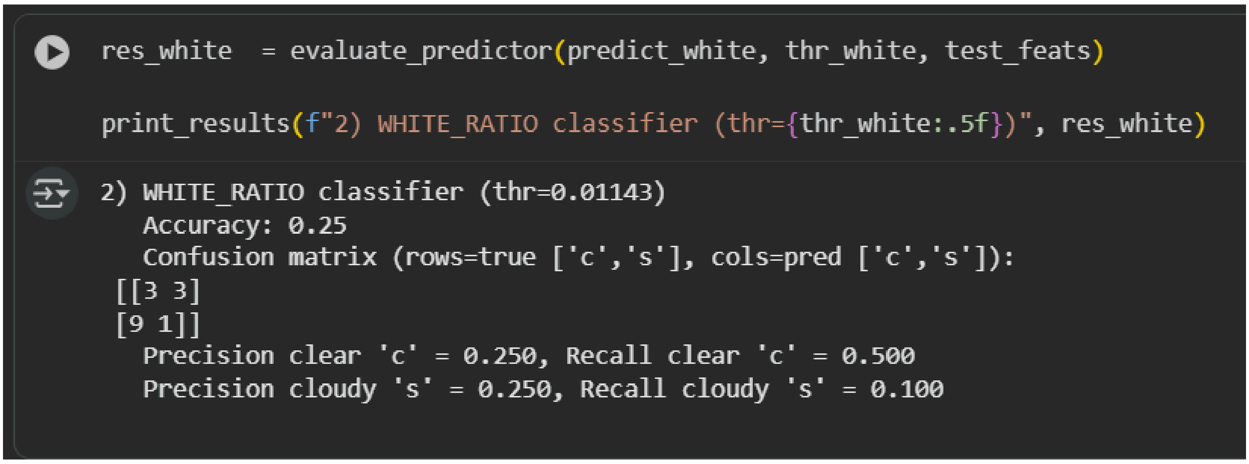
```
res_blue  = evaluate_predictor(predict_blue, thr_blue, test_feats)

print_results(f"2) BLUE_RATIO classifier (thr={thr_blue:.5f})", res_blue)
```

```
2) BLUE_RATIO classifier (thr=126.91335)
   Accuracy: 0.5
   Confusion matrix (rows=true ['c','s'], cols=pred ['c','s']):
 [[3 3]
 [5 5]]
   Precision clear 'c' = 0.375, Recall clear 'c' = 0.500
   Precision cloudy 's' = 0.625, Recall cloudy 's' = 0.500
```

Figure 10. Results of the blue ratio classifier.

3-9. عنوان بخش

With the same reasoning of subsection 1 I will now focus on the color white.

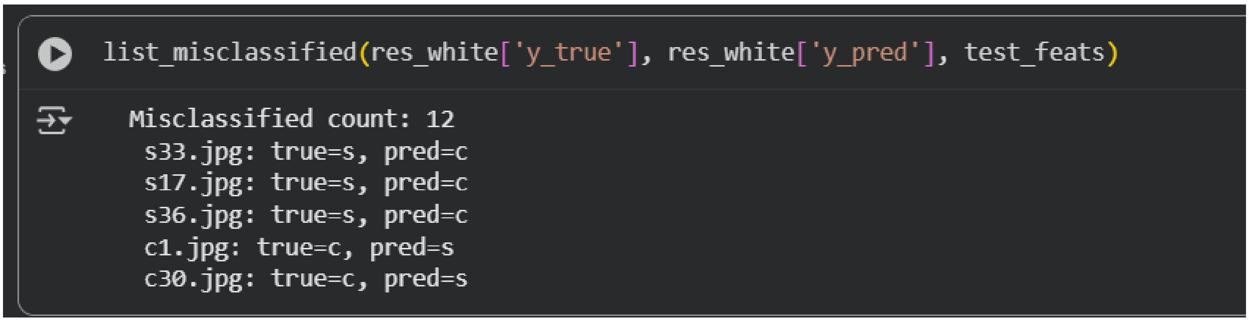


```
res_white  = evaluate_predictor(predict_white, thr_white, test_feats)

print_results(f"2) WHITE_RATIO classifier (thr={thr_white:.5f})", res_white)
```

```
2) WHITE_RATIO classifier (thr=0.01143)
   Accuracy: 0.25
   Confusion matrix (rows=true ['c','s'], cols=pred ['c','s']):
 [[3 3]
 [9 1]]
   Precision clear 'c' = 0.250, Recall clear 'c' = 0.500
   Precision cloudy 's' = 0.250, Recall cloudy 's' = 0.100
```

Figure 11. Results of the white ratio classifier.

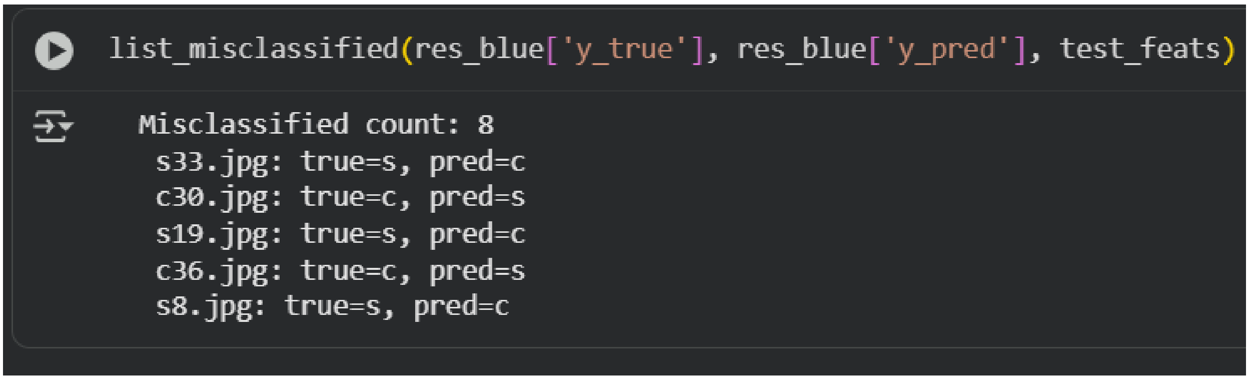
4-9. عنوان بخش



```
list_misclassified(res_white['y_true'], res_white['y_pred'], test_feats)
```

```
Misclassified count: 12
 s33.jpg: true=s, pred=c
 s17.jpg: true=s, pred=c
 s36.jpg: true=s, pred=c
 c1.jpg: true=c, pred=s
 c30.jpg: true=c, pred=s
```

Figure 12. Part 3.



```
list_misclassified(res_blue['y_true'], res_blue['y_pred'], test_feats)
```

```
Misclassified count: 8
 s33.jpg: true=s, pred=c
 c30.jpg: true=c, pred=s
 s19.jpg: true=s, pred=c
 c36.jpg: true=c, pred=s
 s8.jpg: true=s, pred=c
```

Figure 13. Part 1.

Let’s analyze s33 which was misclassified by both.

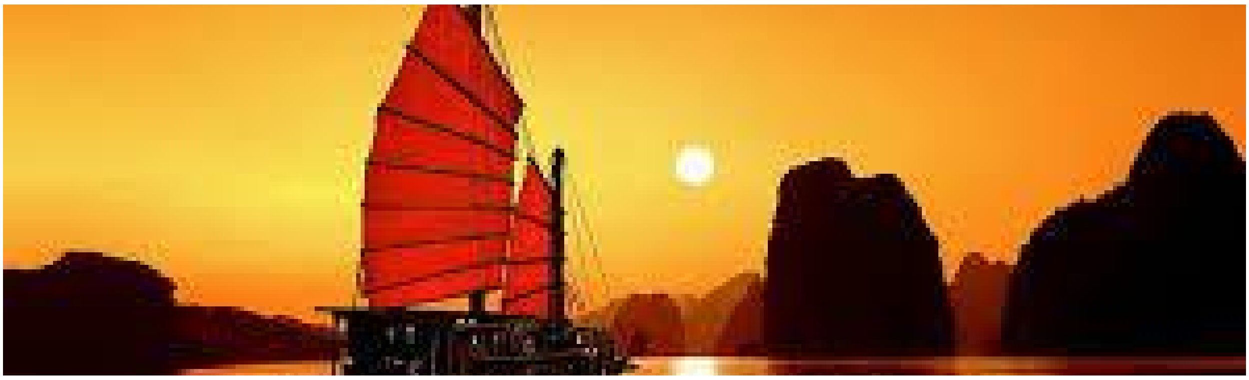


Figure 14.s33.

This image is labeled clear but it has neither a lot of blue pixels nor white pixels! It seems that a clear sky can have more colors and color isn’t a very robust classifier.

9-5. عنوان بخش

As is obvious in Figures 9 and 10 the blue classifier performed somewhat randomly but was slightly better in clear skies, on the other hand the white classifier performed poorly on both specially cloudy skies because clouds can also be white!

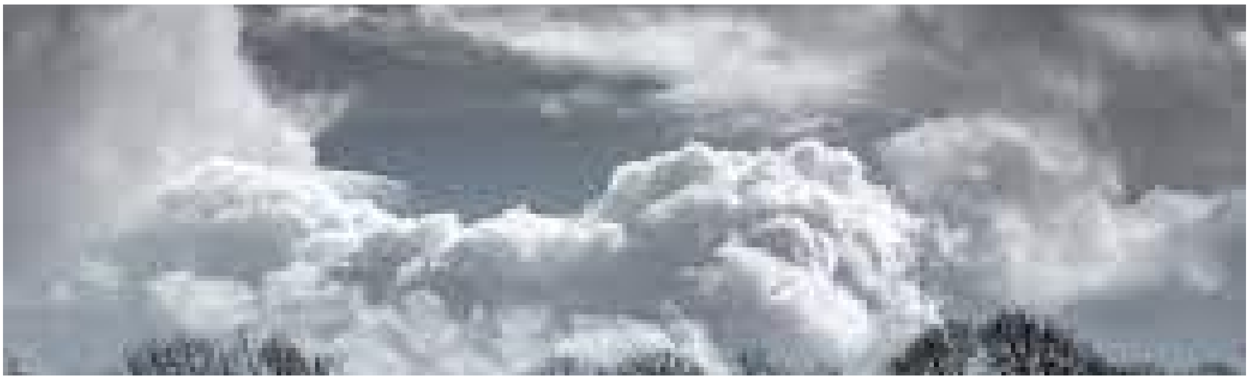


Figure 15.c1 misclassified by white..

This image has a lot of white pixels!

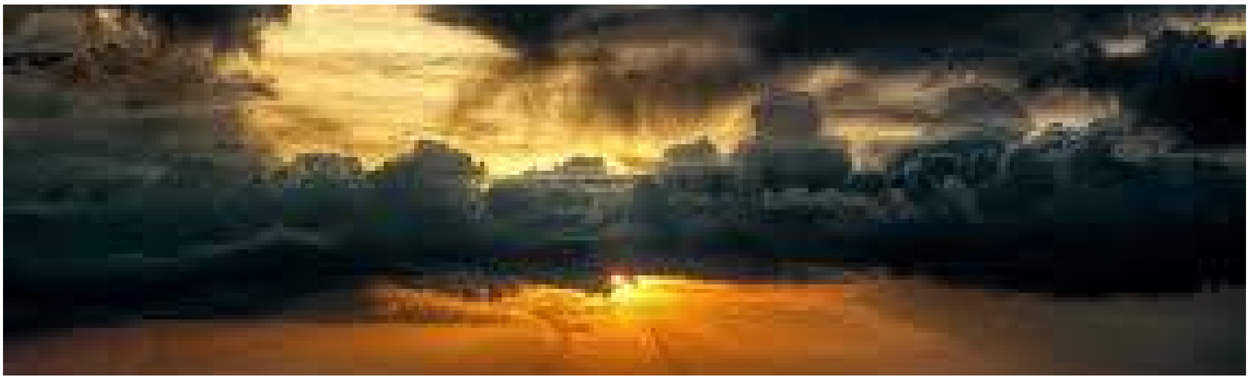


Figure 16.c36 misclassified by blue.

This image too has a lot of blue pixels but it is cloudy!