

## 비정형 데이터 분석 with Orange 3

컴퓨터학부 2014097056 우성현

### 1. 과제 개요

본 과제에서는 이미지 데이터, 텍스트 데이터와 같은 비정형 데이터의 분석을 실습해 본다. 또한, 인공지능망을 이용한 이미지 분류, 텍스트 분석을 통한 스팸 필터 만들기를 통해 머신러닝과 인공지능 알고리즘의 동작 원리를 학습한다.

### 2. MNIST 필기체 인식



#### ► Data set Description

The data files train.csv and test.csv contain gray-scale images of hand-drawn digits, from zero through nine.

Each image is 28 pixels in height and 28 pixels in width, for a total of 784 pixels in total. Each pixel has a single pixel-value associated with it, indicating the lightness or darkness of that pixel, with higher numbers meaning darker. This pixel-value is an integer between 0 and 255, inclusive.

The training data set, (train.csv), has 785 columns. The first column, called "label", is the digit that was drawn by the user. The rest of the columns contain the pixel-values of the associated image.

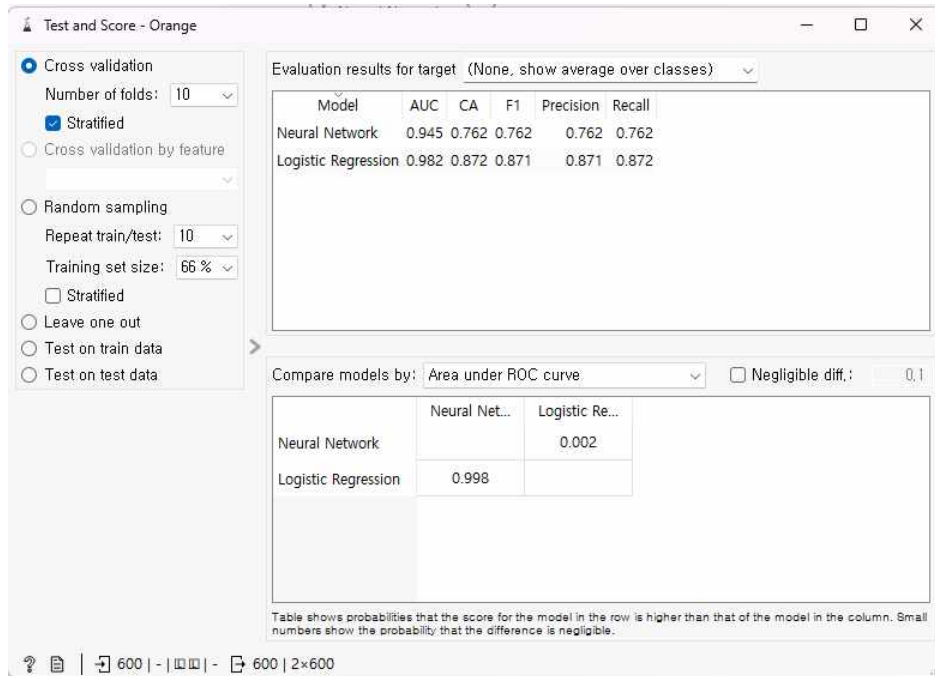
Each pixel column in the training set has a name like pixel<sub>x</sub>, where x is an integer between 0 and 783, inclusive. To locate this pixel on the image, suppose that we have decomposed x as  $x = i * 28 + j$ , where i and j are integers between 0 and 27, inclusive. Then pixel<sub>x</sub> is located on row i and column j of a 28 x 28 matrix, (indexing by zero).

1) MNIST 데이터셋으로 Logistic Regression, Neural Network을 이용한 이미지 분류 실험을 실행한다.

- 위 실험 결과에서 Logistic Regression, Neural Network 모델의 AUC 지표와 F1-score의 값은 각각 얼마인가? 두 모델의 성능에는 차이가 있는가? 있다면, 어느 쪽의 성능이 더 나은가?

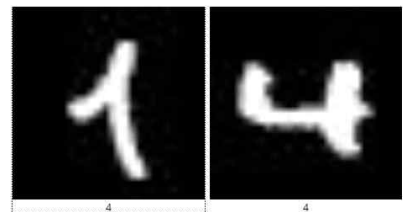
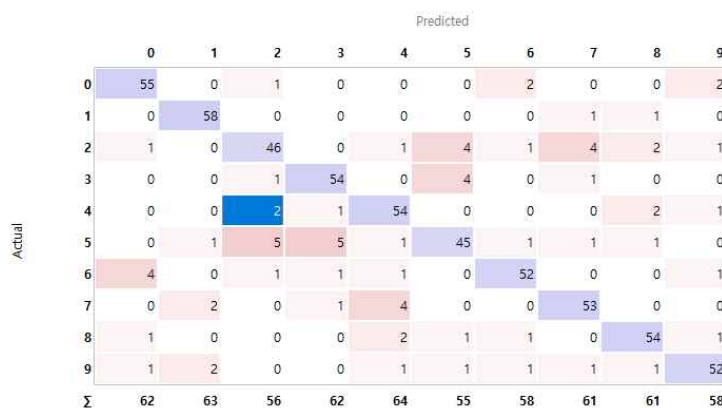
▶ Logistic : 0.982, 0.871 / Neural : 0.945, 0.762

두 모델의 성능 차이는 있으며, Logistic Regression 모델이 모든 지표에서 더 좋은 성능을 보여주고 있다.



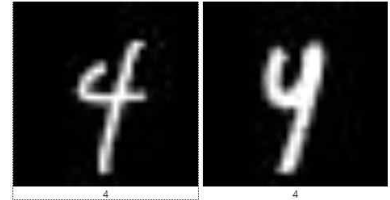
2) 위 1번 실험의 결과로 나온 Confusion Matrix 화면을 분석하시오.

- Logistic Regression의 Confusion Matrix를 캡처하여 제시하고, Logistic Regression이 오분류한 이미지를 두 개 이상 제시하시오.



- Neural Network의 Confusion Matrix를 캡처하여 제시하고, Neural Network이 오분류한 이미지를 두 개 이상 제시하시오.

| Actual \ Predicted |    |    |    |    |    |    |    |    |    |    |
|--------------------|----|----|----|----|----|----|----|----|----|----|
|                    | 0  | 1  | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  |
| 0                  | 50 | 0  | 1  | 0  | 1  | 1  | 4  | 0  | 0  | 3  |
| 1                  | 0  | 57 | 1  | 0  | 1  | 0  | 0  | 1  | 0  | 0  |
| 2                  | 0  | 0  | 38 | 1  | 2  | 6  | 3  | 3  | 6  | 1  |
| 3                  | 0  | 0  | 4  | 44 | 0  | 8  | 0  | 0  | 1  | 3  |
| 4                  | 0  | 2  | 3  | 1  | 42 | 3  | 0  | 5  | 3  | 1  |
| 5                  | 1  | 0  | 6  | 9  | 3  | 39 | 1  | 0  | 1  | 0  |
| 6                  | 6  | 0  | 2  | 1  | 1  | 2  | 45 | 0  | 1  | 2  |
| 7                  | 0  | 4  | 2  | 0  | 3  | 1  | 1  | 47 | 0  | 2  |
| 8                  | 0  | 0  | 2  | 0  | 2  | 1  | 1  | 0  | 50 | 4  |
| 9                  | 1  | 1  | 1  | 2  | 3  | 2  | 1  | 1  | 3  | 45 |
| Σ                  | 58 | 64 | 60 | 58 | 58 | 63 | 56 | 57 | 65 | 61 |



- ▶ Logistic Regression :  $523/600 = 0.872$  (CA)
- ▶ Neural Network :  $457/600 = 0.762$  (CA)

3) 위의 설정에서 Neurons in hidden layers를 변경했을 때, Test and Score 화면을 분석하시오.

- Neurons in hidden layers를 10에서 100으로 변경하면 분류기의 성능에 어떤 변화가 일어나는가?

| Model          | AUC   | CA    | F1    | Precision | Recall |
|----------------|-------|-------|-------|-----------|--------|
| Neural Network | 0.945 | 0.762 | 0.762 | 0.762     | 0.762  |
| Neural Network | 0.984 | 0.842 | 0.841 | 0.841     | 0.842  |

- ▶ hidden layers의 option 값을 100으로 변경 후, AUC를 비롯해 전반적으로 높은 점수를 보여준다.

- Neurons in hidden layers를 10, 10, 10으로 변경하면 분류기의 성능에 어떤 변화가 일어나는가?

| Model          | AUC   | CA    | F1    | Precision | Recall |
|----------------|-------|-------|-------|-----------|--------|
| Neural Network | 0.945 | 0.762 | 0.762 | 0.762     | 0.762  |
| Neural Network | 0.882 | 0.588 | 0.560 | 0.553     | 0.588  |

- ▶ hidden node를 10개씩 3개의 층을 쌓은 모델이 오히려 10개의 노드(Pesceptron)만 사용한 모델보다 더 낮게 나왔다.

- Neurons in hidden layers를 다양하게 변화하는 실험을 하고, 은닉층을 몇 층으로 쌓는가, 은닉층의 퍼셉트론 갯수를 어떻게 지정하는가(동일하게, 증가하도록, 감소하도록)에 따라 분류기의 성능에 미치는 영향에 대해서 서술하시오.

\*\* test 기준 모델(다른 옵션 동일)

- 1) hidden layer 1개 층의 10개 퍼셉트론으로 이뤄진 모델(다른 옵션 동일)
- 2) hidden layer 1개 층의 100개 퍼셉트론으로 이뤄진 모델

① Neurons in hidden layers : 5 → rank: / 성능 감소

| Model          | AUC   | CA    | F1    | Precision | Recall |
|----------------|-------|-------|-------|-----------|--------|
| Neural Network | 0.945 | 0.762 | 0.762 | 0.762     | 0.762  |
| Neural Network | 0.984 | 0.842 | 0.841 | 0.841     | 0.842  |
| Neural Network | 0.911 | 0.680 | 0.680 | 0.689     | 0.680  |

▶ hidden node의 수를 10개에서 5개로 줄였더니 전반적인 성능이 대폭 감소했다.

② Neurons in hidden layers : 10, 10 → rank: / 성능 감소

| Model          | AUC   | CA    | F1    | Precision | Recall |
|----------------|-------|-------|-------|-----------|--------|
| Neural Network | 0.945 | 0.762 | 0.762 | 0.762     | 0.762  |
| Neural Network | 0.984 | 0.842 | 0.841 | 0.841     | 0.842  |
| Neural Network | 0.929 | 0.695 | 0.696 | 0.699     | 0.695  |

▶ hidden layers를 2개 층으로 쌓았더니 오히려 성능이 감소했다.

③ Neurons in hidden layers : 30, 50, 100 → rank: / 성능 감소

| Model          | AUC   | CA    | F1    | Precision | Recall |
|----------------|-------|-------|-------|-----------|--------|
| Neural Network | 0.945 | 0.762 | 0.762 | 0.762     | 0.762  |
| Neural Network | 0.984 | 0.842 | 0.841 | 0.841     | 0.842  |
| Neural Network | 0.927 | 0.668 | 0.667 | 0.666     | 0.668  |

▶ hidden layers의 수를 3개로, 각 층의 node의 수를 30, 50, 100으로 설정했을 때, layer와 node를 더 적게 사용한 test-model② 보다 성능이 좋지 않았다.

④ Neurons in hidden layers : 100, 100, 100

| Model          | AUC   | CA    | F1    | Precision | Recall |
|----------------|-------|-------|-------|-----------|--------|
| Neural Network | 0.945 | 0.762 | 0.762 | 0.762     | 0.762  |
| Neural Network | 0.984 | 0.842 | 0.841 | 0.841     | 0.842  |
| Neural Network | 0.941 | 0.710 | 0.709 | 0.710     | 0.710  |

▶ 이 모델의 경우 AUC 지표는 첫 번째 기준 모델인 node를 10개 쓴 것과 비교해 비슷하게 나왔지만, 다른 지표는 오히려 좋지 않았다.

⑤ Neurons in hidden layers : 150

| Model          | AUC   | CA    | F1    | Precision | Recall |
|----------------|-------|-------|-------|-----------|--------|
| Neural Network | 0.945 | 0.762 | 0.762 | 0.762     | 0.762  |
| Neural Network | 0.984 | 0.842 | 0.841 | 0.841     | 0.842  |
| Neural Network | 0.984 | 0.828 | 0.828 | 0.828     | 0.828  |

▶ 첫 번째 기준 모델보다는 좋은 성능이 나왔지만, 단층의 100개 node를 사용한 두 번째 기준 모델보다는 좋지 않게 나왔다.

**\*\* test 결론**

▶ 위의 실험 결과,

- 1) 단층의 hidden layer를 사용하는 것이 가장 우수한 성능을 보여주었다.
- 2) layer의 node는 100개를 설정했을 때 가장 우수한 성능을 보여주었다.

4) 위 3)번 실험에서 Activation Function을 변경했을 때, Test and Score 화면을 분석하시오.

- 다음 각 활성 함수의 성능을 비교해 본 Test and Score 화면을 제시하고, 활성 함수의 어떤 성질이 결과에 영향을 미치는가를 서술하시오.

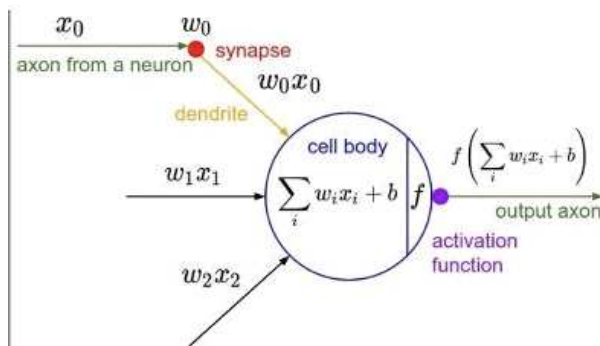
Activation: ①Identity, ②Logistic, ③tanh, ④ReLU

- test 모델 : 3)번 실험에서 가장 우수한 모델인 Neurons in hidden layers : 100(다른 조건 동일)

\*\* Test and Score

|   | Model          | AUC   | CA    | F1    | Precision | Recall |
|---|----------------|-------|-------|-------|-----------|--------|
| ① | Neural Network | 0.986 | 0.862 | 0.862 | 0.863     | 0.862  |
| ② | Neural Network | 0.984 | 0.842 | 0.841 | 0.841     | 0.842  |
| ③ | Neural Network | 0.988 | 0.872 | 0.871 | 0.872     | 0.872  |
| ④ | Neural Network | 0.987 | 0.862 | 0.861 | 0.862     | 0.862  |

- ▶ Test 결과 전반적으로 비슷한 수치를 보이지만, 활성 함수를 tanh로 사용한 test model-③이 가장 좋은 성능을 보여주고 있다.



(출처 : <https://saintbinary.tistory.com/8> )

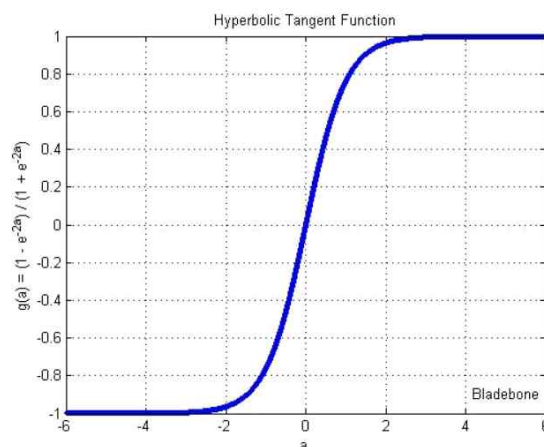
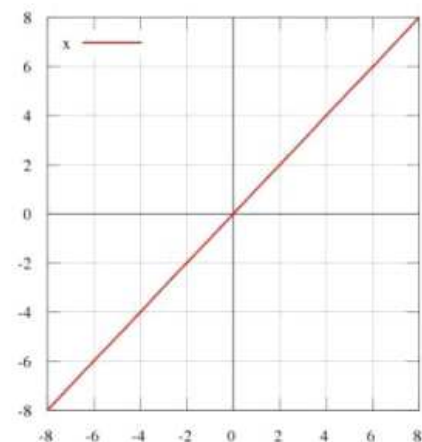
-> 활성 함수는 Neural Network에서 입력받은 데이터를 다음 층으로 출력할지 결정하는 역할을 한다.

-> 각 함수마다 출력하는 수식이 다른데, 해당 MNIST 손글씨를 분류하는데 tanh 활성화 함수가 가장 잘 맞기에 좋은 성능을 보여준다.

\*\* activation function

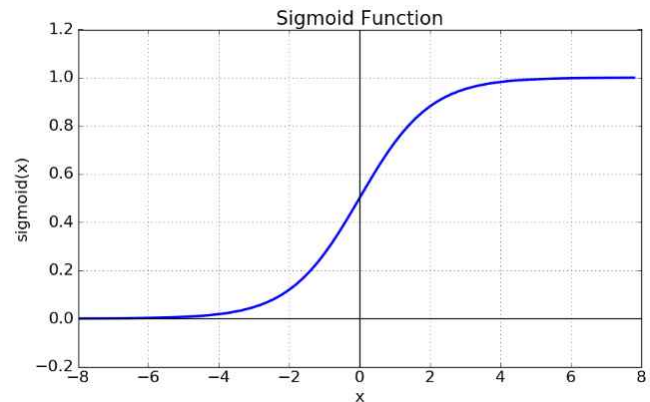
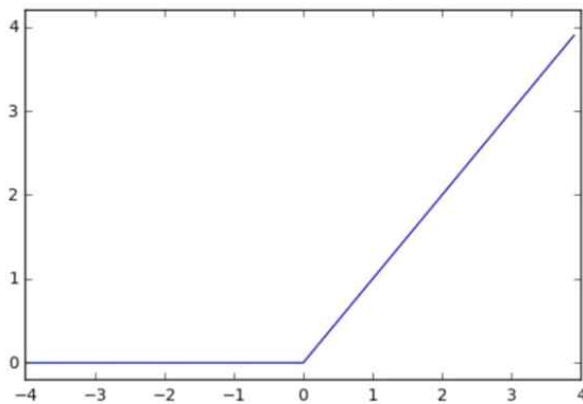
- Identity : 항등함수로, 입력값의 가중합이 그대로 출력된다.

- Tanh : Sigmoid Function을 재할용해  $[-1, 1]$ 의 범위를 갖는 함수이다.





- **ReLU** : Gradient Vanishing 문제를 해결하기 위한 함수이다. (x값이 음수일 경우 0을 출력)
- **Logistic** : sigmoid 함수라고도 하며 함수 값은  $[0, 1]$ 이고, 중간값은  $1/2$ 이다.



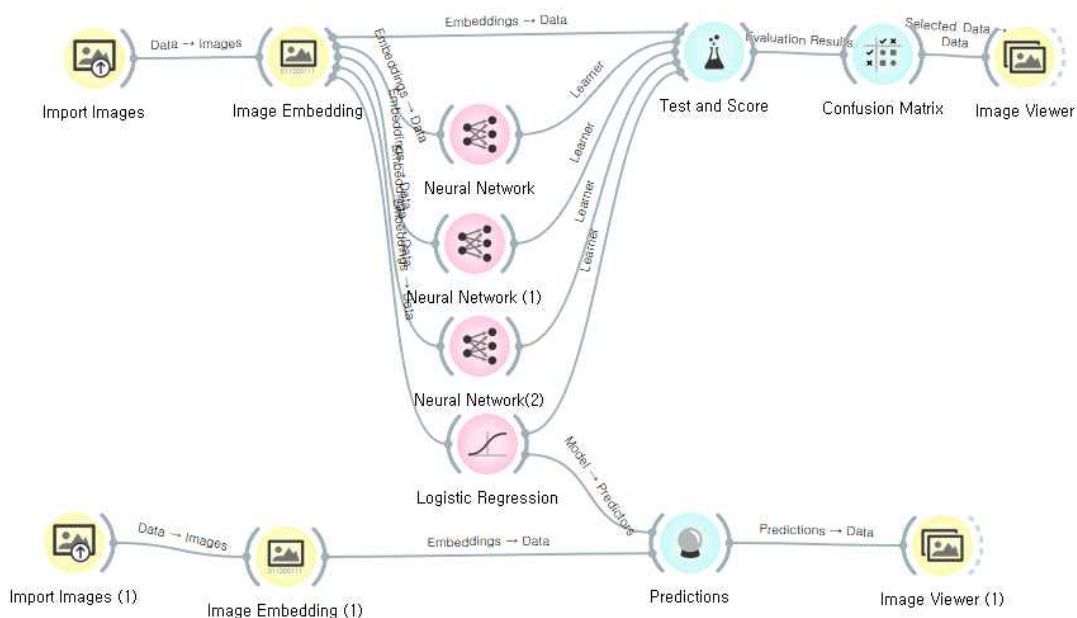
5) 위 3)번 실험에서 Solver를 변경했을 때, Test and Score 화면을 분석하시오.

- 다음 각 Solver의 성능을 비교해 본 Test and Score 화면을 제시하시오.

Solver: ①L-BFGS-B, ②SGD, ③Adam

|   | Model          | AUC   | CA    | F1    | Precision | Recall |
|---|----------------|-------|-------|-------|-----------|--------|
| ① | Neural Network | 0.988 | 0.872 | 0.871 | 0.872     | 0.872  |
| ② | Neural Network | 0.984 | 0.843 | 0.842 | 0.843     | 0.843  |
| ③ | Neural Network | 0.988 | 0.882 | 0.881 | 0.882     | 0.882  |

▶ Solver의 성능을 비교했더니 경사(관성)에 따라 step size를 다르게 하는③Adaptive Moment Estimation 학습 방법이 가장 높은 성능을 기록했다. 그렇지만, 나머지 학습 방법들도 크게 차이가 나지 않으며 모두 높은 성능을 보여주고 있다.



### 3. 스팸 메일 분류기

## SMS Spam Collection Dataset

Collection of SMS messages tagged as spam or legitimate



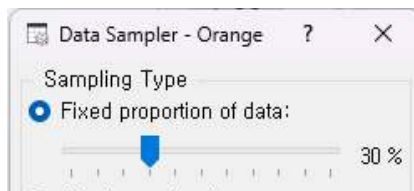
#### ► Data set Description

The SMS Spam Collection is a set of SMS tagged messages that have been collected for SMS Spam research. It contains one set of SMS messages in English of 5,574 messages, tagged according to being ham (legitimate) or spam.

1) SMS spam 데이터셋으로 Logistic Regression, Naive Bayesian, Neural Network을 이용한 텍스트 분류 실험을 실행한다.

- 위 실험 결과에서 Logistic Regression, Neural Network 모델의 AUC 지표와 F1-score의 값은 각각 얼마인가? 두 모델의 성능에는 차이가 있는가? 있다면, 어느 모델의 성능이 더 나은가?

\*\* test 조건: 학습에 많은 시간이 소요돼 Data Sampler Widget을 이용해 30%의 데이터로 진행.

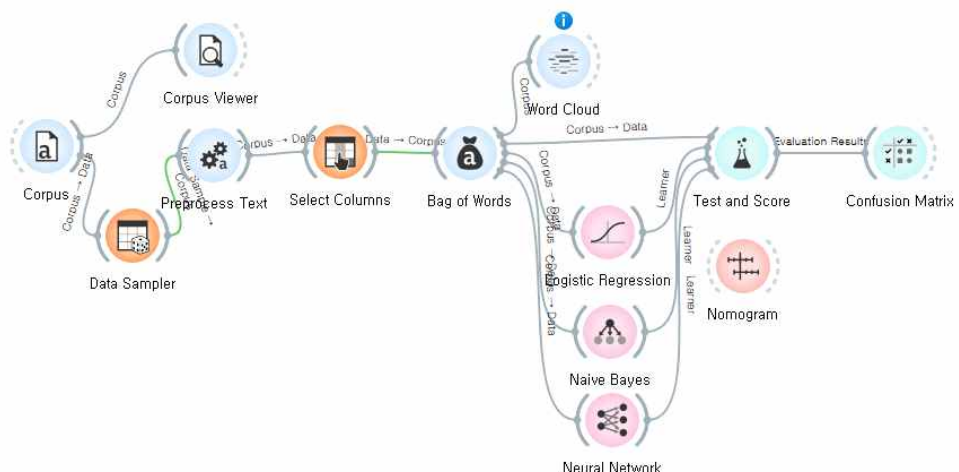


| Model               | AUC   | CA    | F1    | Precision | Recall |
|---------------------|-------|-------|-------|-----------|--------|
| Neural Network      | 0.984 | 0.983 | 0.982 | 0.983     | 0.983  |
| Logistic Regression | 0.970 | 0.974 | 0.973 | 0.974     | 0.974  |

- Logistic Regression : 0.970 / 0.973

- Neural Network : 0.984 / 0.982

► test 결과 : Neural Network 모델이 AUC: +0.014, F1: +0.009 만큼 조금 더 좋은 성능을 보여주고 있다.







3) 위 1번 실험의 결과로 나온 Confusion Matrix 화면을 분석하시오.

- Logistic Regression의 Confusion Matrix를 캡처하여 제시하고, Logistic Regression이 오분류한 텍스트를 두 개 이상 제시하시오.

|          |      | Predicted |      |          |
|----------|------|-----------|------|----------|
|          |      | ham       | spam | $\Sigma$ |
| Actual   | ham  | 1447      | 10   | 1457     |
|          | spam | 61        | 150  | 211      |
| $\Sigma$ |      | 1508      | 160  | 1668     |

#### ① ham -> spam 오분류 문장 :10

- 'U free on sat rite? U wan 2 watch infernal affairs wif me n darren n mayb xy?'
- 'Call me when u finish then i come n pick u.'

#### ② spam -> ham 오분류 문장 : 61

- 'FREE>Ringtone!Reply REAL'
- 'Got what it takes 2 take part in the WRC Rally in Oz? U can with Lucozade Energy! Text RALLY LE to 61200 (25p) see packs or lucozade.co.uk/wrc & itcould be u!'

#### ► Confusion Matrix 결과

ham mail을 spam으로 오분류 한 것보다, spam 메일을 ham 메일로 오분류한 것이 6배가량 더 많다. Logistic Regression의 회귀계수를 보면 '£ (6.24768)', 'Call(4.69928)', 'FREE(3.58894)' 등이 높은데 ham을 spam이라고 오분류한 문장들의 Bag of Words를 보면 이러한 단어들이 포함되어 있음을 알 수 있다.

| name      | spam      |
|-----------|-----------|
| 96 £      | 6.24768   |
| 95 www    | 4.71029   |
| 5 Call    | 4.69928   |
| 20 To     | 3.8052    |
| 7 FREE    | 3.58894   |
| 58 mobile | 2.85      |
| 85 text   | 2.10727   |
| 78 send   | 1.42762   |
| 70 our    | 1.25683   |
| 90 ur     | 1.2553    |
| 33 call   | 1.13985   |
| 41 free   | 1.08138   |
| 77 see    | 1.0156    |
| 71 phone  | 0.979322  |
| 32 been   | 0.925413  |
| 21 U      | 0.827201  |
| 25 Your   | 0.825603  |
| 27 an     | 0.819272  |
| 94 what   | 0.726445  |
| 24 You    | 0.60735   |
| 87 time   | 0.440256  |
| 69 only   | 0.251772  |
| 28 any    | 0.175845  |
| 29 as     | 0.0296112 |

| type | text (1)            | Logistic Regres: | {...}                                     |
|------|---------------------|------------------|---|
|      | True                |                  |   |
| ham  | U free on sat ri... | spam             | U=2, free=1, n=2                          |
| ham  | Call me when ...    | spam             | Call=1, come=1, n=1, pick=1               |
| ham  | Aight I've been...  | spam             | It=1, as=2, been=1, doing=1, free=1, t... |
| ham  | Hi Mobile no. ...   | spam             | #=1, Hi=1, It=1, free=1, great=1, has=... |
| ham  | Can you call m...   | spam             | Your=1, call=3, number=1                  |
| ham  | Your account h...   | spam             | #=1, Your=3, been=1, has=1                |
| ham  | HELLOGORGEO...      | spam             | U=4                                       |
| ham  | Plz note: if any... | spam             | #=4, Do=1, an=1, been=1, has=1, mak...    |
| ham  | Hi Mobile no. ...   | spam             | #=1, Hi=1, It=1, free=1, great=1, has=... |
| ham  | Your account h...   | spam             | #=1, Your=3, been=1, has=1                |

- Neural Network의 Confusion Matrix를 캡처하여 제시하고, Neural Network이 오분류한 텍스트를 두 개 이상 제시하시오.

|        |      | Predicted |      |      |
|--------|------|-----------|------|------|
|        |      | ham       | spam | Σ    |
| Actual | ham  | 1429      | 28   | 1457 |
|        | spam | 43        | 168  | 211  |
| Σ      |      | 1472      | 196  | 1668 |

#### ① ham -> spam 오분류 문장 : 28

- 'I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times.'
- 'Send ur birthdate with month and year I will tel u ur LIFE PARTNER'S name. and the method of calculation. Reply must.'

#### ② spam -> ham 오분류 문장 : 43

- 'Rock yr chik. Get 100's of filthy films &XXX pics on yr phone now. rply FILTH to 69669. Saristar Ltd E14 9YT 08701752560. 450p per 5 days. Stop2 cancel'
- '07732584351 - Rodger Burns - MSG = We tried to call you re your reply to our sms for a free nokia mobile + free camcorder. Please call now 08000930705 for delivery tomorrow'

#### ► Confusion Matrix 결과

Neural Network 모델과 Logistic 모델의 오분류 수는 71개로 동일하다. 그렇지만, spam을 ham으로 오분류한 수는 Neural Network의 수가 더 적다. 즉 더 많은 스팸을 분류해냈다고 할 수 있다. spam 분류의 특성을 고려해볼 때, 오분류의 수가 동일하다면 Logistic 모델을 사용하는 것이 더 유리하다. 왜냐하면 ham을 spam이라고 분류해 사용자에게 보여주지 않는다면 치명적인 문제를 가져올 수 있기 때문이다.

| type | text (1)           | ie(Neural Netwo | {...}                            |
|------|--------------------|-----------------|----------------------------------|
|      | True               |                 |                                  |
| ham  | I've been searc... | spam            | You=1, been=2, take=1, ve=1      |
| ham  | Send ur birthd...  | spam            | ur=2                             |
| ham  | Hello! How r u...  | spam            | has=1, r=1                       |
| ham  | Can... I'm free... | spam            | free=1                           |
| ham  | Just seeing you... | spam            | Do=1, Just=1, call=1, day=1      |
| ham  | Cos daddy arra...  | spam            | time=2                           |
| ham  | When i have st...  | spam            | ll=1, tell=1                     |
| ham  | This is my num...  | spam            | number=1                         |
| ham  | Call me when ...   | spam            | Call=1, come=1, n=1, pick=1      |
| ham  | Even u dont ge...  | spam            | dont=3                           |
| ham  | Somewhr som...     | spam            | And=1, has=1, make=1, time=1,... |
| ham  | I have 2 sleepi... | spam            | phone=1                          |
| ham  | Stupid auto co...  | spam            | phone=1                          |
| ham  | Oh shut it. Om...  | spam            | about=1, only=1                  |
| ham  | Today i'm not ...  | spam            | free=1, ur=1                     |

- Naive Bayes의 Confusion Matrix를 캡처하여 제시하고, Naive Bayes가 오분류한 텍스트를 두 개 이상 제시하시오.

|          |      | Predicted |      |          |
|----------|------|-----------|------|----------|
|          |      | ham       | spam | $\Sigma$ |
| Actual   | ham  | 1434      | 23   | 1457     |
|          | spam | 51        | 160  | 211      |
| $\Sigma$ |      | 1485      | 183  | 1668     |

#### ① ham -> spam 오분류 문장 : 23

- 'Aiyar hard 2 type. U later free then tell me then i call n scold n tell u.'
- 'U free on sat rite? U wan 2 watch infernal affairs wif me n darren n mayb xy?'

#### ② spam -> ham 오분류 문장 : 51

- 'Rock yr chik. Get 100's of filthy films &XXX pics on yr phone now. rply FILTH to 69669. Saristar Ltd E14 9YT 08701752560. 450p per 5 days. Stop2 cancel'
- 'SIX chances to win CASH! From 100 to 20000 pounds txt> CSH11 and send to 87575. Cost 150p/day 6days 16+ TsandCs apply Reply HL 4 info'

#### ► Confusion Matrix 결과

Naive Bayes 모델의 총 오분류 문장의 수는 74로 다른 두 모델과 비교했을 때, 3문장 정도 더 많다.

| bow-feature<br>hidden<br>skip-normalization<br>title | type | text (1)            | type(Naive Bayes) | {...}   |
|--|------|---------------------|-------------------|---|
|  |      | True                |                   |   |
| 1  | ham  | Aiyar hard 2 ty...  | spam              | U=1, call=1, free=1, later=1, n=2, tell=2       |
| 2  | ham  | U free on sat ri... | spam              | U=2, free=1, n=2                                |
| 3  | ham  | Hi Harish's rent... | spam              | Hi=1, been=1, has=1, ur=1                       |
| 4  | ham  | Hi baby ive jus...  | spam              | Hi=1, lf=1, back=1, call=1, got=1, phone=1,...  |
| 5  | ham  | NEFT Transacti...   | spam              | #=3, been=1, has=1, number=1                    |
| 6  | ham  | Once free call ...  | spam              | call=1, free=1                                  |
| 7  | ham  | Aight I've been...  | spam              | lt=1, as=2, been=1, doing=1, free=1, text=1,... |
| 8  | ham  | Aft i finish my ... | spam              | U=1, go=1, lor=2, ur=1                          |
| 9  | ham  | Mum hope yo...      | spam              | Have=1, day=2, great=2, text=1                  |
| 10   | ham  | Hi Mobile no. ...   | spam              | #=1, Hi=1, lt=1, free=1, great=1, has=1, mo...  |
| 11   | ham  | Today i'm not ...   | spam              | free=1, ur=1                                    |
| 12   | ham  | Can you call m...   | spam              | Your=1, call=3, number=1                        |
| 13   | ham  | Take care and s...  | spam              | Your=1, more=1, need=2, no=2, only=1, tim...    |

- 위 세 개의 분류기의 Confusion Matrix를 비교했을 때, 특이한 점을 발견하고, 그 의미에 대해서 설명해 보시오.

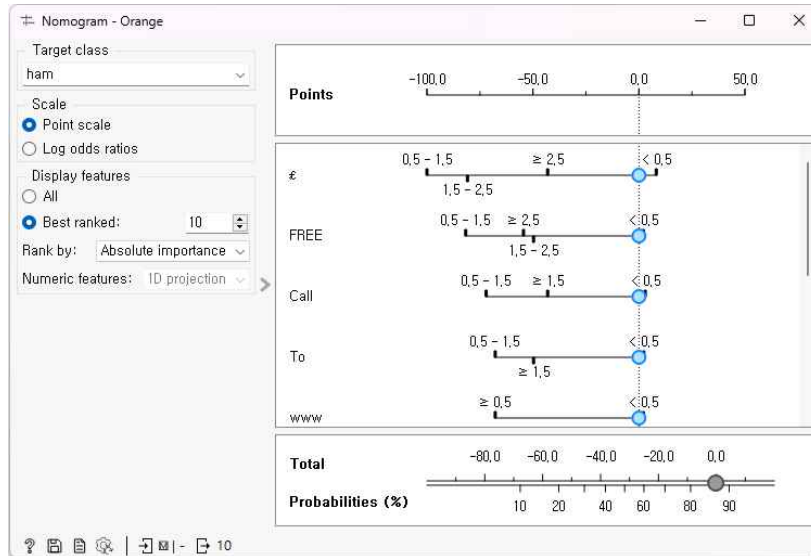
►랜덤 샘플링으로 30%의 데이터만 사용해서 그런지 크게 특이하다고 생각되는 부분이 없었지만, 세 분류기 모두 비슷한 성능을 보여준다는 것과 Neural Network 모델을 분석할 때 언급 했듯이 분류기마다 더 잘 분류하는 파트가 다르다는 것이다.

4) 위 1번 실험의 결과로 나온 Logistic Regression에 연결된 Nomogram을 분석하시오.

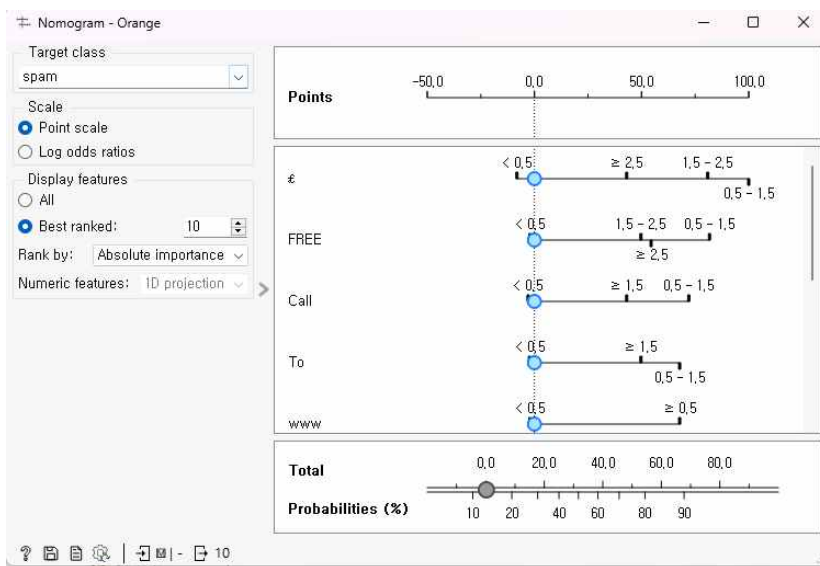
- Nomogram의 Target class를 바꿔서 그린 화면을 캡처하여 제시하고, 그 차이에 대해서 의미를 설명하시오.

\*\* Logistic Regression에 연결했더니 target class에 아무 값도 나오지 않아 Naive Bayes 모델을 통해서 진행했습니다. 방법을 찾으려고 구글링을 해봤지만 해결책을 찾지 못했습니다.

#### ① Target class : ham



#### ② Target class : spam



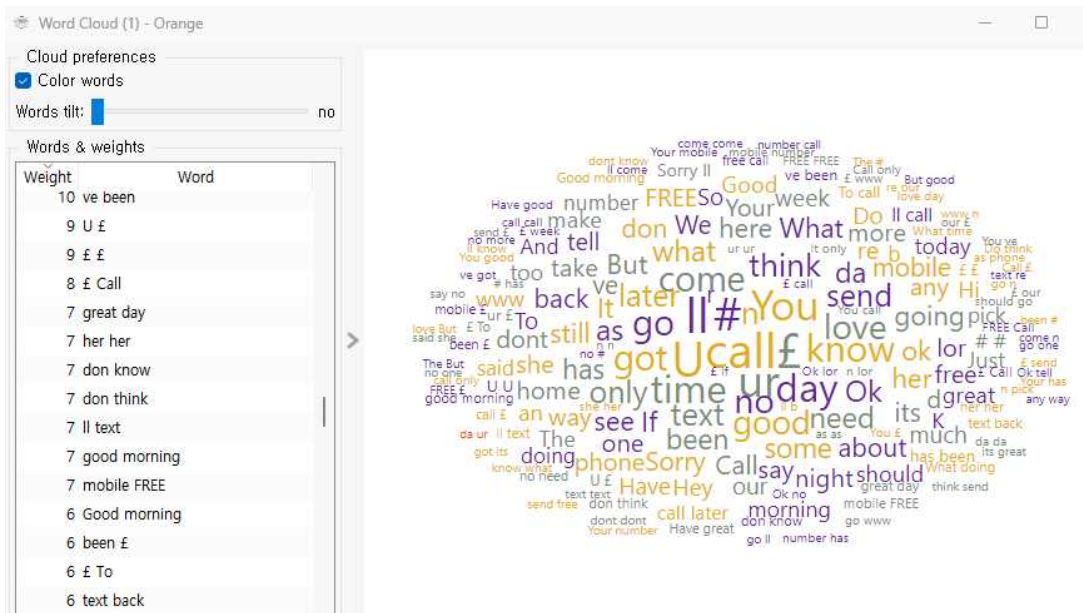
▶ nomogram은 Naive Bayes와 Logistic model의 시각적 표현을 가능하게 한다. 그리고 train data의 구조와 class 확률에 대한 속성의 영향에 대한 통찰력을 제공한다. Rank by Absolute importance로 설정했을 때, '£', 'FREE', 'Call', 'To', 'www' 등이 spam을 분류하는데 가장 중요한 단어라는 것을 알 수 있다. 이때 Target class를 ham으로 하는 경우 파란색 점들이 우측에 있고, Total의 확률도 90%에 가까이 있다. 우측에 있다는 것은 0.5보다 작은 경우인데 이 값이 좌측으로 갈수록 2.5보다 커지고 이와 함께 Total의 Probabilities도 감소한다. 즉, 위의 단어들이 많이 나올수록 ham일 확률이 감소한다는 것을 의미한다. Target class가 spam인 경우는 반대의 결과를 보여준다.

5) 위 1번 실험에서 다양한 자연어 처리 실험을 해보시오.

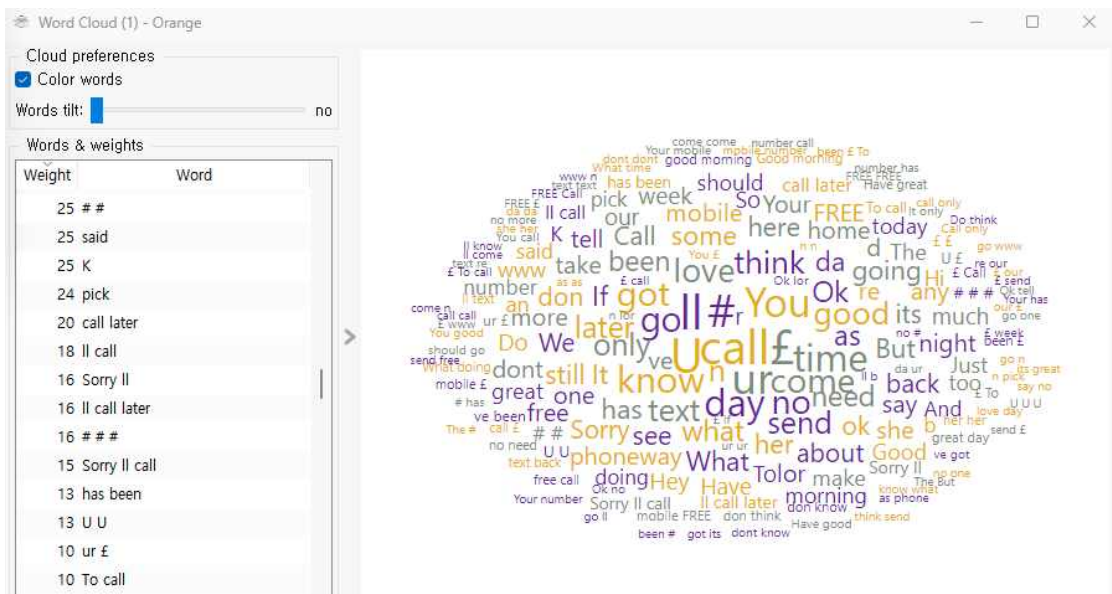
- Preprocess Text에서 N-grams Range를 적용하고, Range를 다양하게 바꿔보고, 옵션을 바꿨을 때의 전처리 결과에 대해서 설명해 보시오.

▶ 한 단어 뿐 아니라 2개의 단어가 묶인 Word를 확인할 수 있다. 2개의 단어로 묶인 2-grams에서는 '# #', 'call later' 등이, 3개의 단어가 묶인 3-grams에서는 'll call later', 'Sorry ll call' 등이 보이는데, 'll'는 아마 토큰화하는 과정에서 'l'이 떼어지고 남은 부분으로 생각된다.

#### ① N-Grams Range : 1~2



#### ② N-Grams Range : 1~3

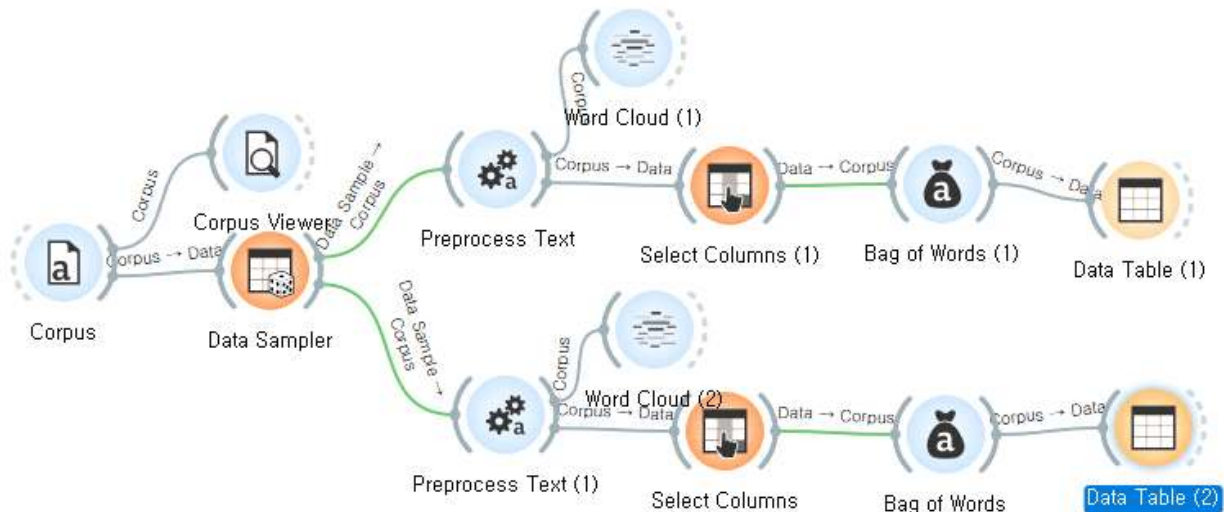




- Preprocess Text에서 Filtering의 POS tags(Noun, Verb)를 적용해 보고, 옵션을 바꿨을 때의 전처리 결과에 대해서 설명해 보시오.

: POS(Part-Of-Speech) tagger는 각 단어의 품사를 붙이는 절차이다. 품사를 알아야 단어 의미의 중의성을 해소할 수 있어 Text mining전 반드시 거쳐야 하는 전처리 과정이다.

POS tags Filtering을 적용한 변화를 식별하기 위해서 아래와 같이 구성해서 테스트를 진행했으나, Word Cloud와 Data Table 모두 동일한 결과 값이 나와서 이 data set에서는 유의미한 차이를 확인하지 못했다.



| Data Table (2) - Orange   |        |                    |     | Data Table (1) |                      |   |   |
|---|--------|--------------------|-----|----------------|----------------------|---|---|
| <div>Info</div> <div>1688 instances</div> <div>3250 features (sparse, density 0.34 %)</div> <div>Target with 2 values</div> <div>1 meta attribute</div> <div>Variables</div> <div>Show variable labels (if present)</div> <div>Visualize numeric values</div> <div>Color by instance classes</div> <div>Selection</div> <div>Select full rows</div> |        |                    |     |                |                      |   |   |
| bow-feature   | hidden | skip-normalization | ... | type           | text (1)             | ...   | ...   |
| 1   |        |                    |     | ham            | He says hi and...    | a=1, and=1, and to=1, at=1, at a=1, back=1, back to=1, get=1, get b...          | a=1, and=1, and to=1, at=1, at a=1, back=1, back to=1, g...               |
| 2   |        |                    |     | ham            | It certainly puts... | It=1, it when=1, like=1, like this=1, this=1, when=1, when...                   | It=1, it when=1, like=1, like this=1, this=1, when=1, when...             |
| 3   |        |                    |     | ham            | Why you Dint...      | come=1, come with=1, with=1, you=1, you come=1                                  | come=1, come with=1, with=1, you=1, you come=1                            |
| 4   |        |                    |     | ham            | at bruce b dow...    | at=1, at now=1, now=1   | at=1, at now=1, now=1   |
| 5   |        |                    |     | ham            | ... Are you in t...  | in=1, in the=1, the=1, you=1, you in=1  | in=1, in the=1, the=1, you=1, you in=1                                    |
| 6   |        |                    |     | ham            | Then u wait 4...     | if=1, if is=1, at=1, at ur=1, back=1, back n=1, come=1, come back=1...          | if=1, if is=1, at=1, at ur=1, back=1, back n=1, come=1, come back=1...    |
| 7   |        |                    |     | ham            | Nt joking serio...   | is=1  | is=1  |
| 8   |        |                    |     | ham            | Ok but tell me...    | Ok=1, Ok but=1, but=1, but me=1, come=1, come is=1, is=1, I need=1...           | Ok=1, Ok but=1, but=1, but me=1, come=1, come is=1, is=1, I need=1...     |
| 9   |        |                    |     | ham            | LMAO where's...      | I=1, I need=1, it=1, need=1, need it=1, s=1, s your=1, when=1, when...          | I=1, I need=1, it=1, need=1, need it=1, s=1, s your=1, when=1, when...    |
| 10  |        |                    |     | ham            | All boys made...     | I=1, I just=1, Ok=1, Ok is=1, for=1, have=1, have no=1, is=1, I have=1...       | I=1, I just=1, Ok=1, Ok is=1, for=1, have=1, have no=1, is=1, I have=1... |
| 11  |        |                    |     | ham            | Tunji how's the...   | a=1, a day=1, are=1, are you=1, day=1, how=2, how are=1, how s=1...             | a=1, a day=1, are=1, are you=1, day=1, how=2, how are=1, how s=1...       |
| 12  |        |                    |     | ham            | I'm home. Doc...     | I=1, I m=1, is=1, m=1, m me=1, me=1, me is=1                                    | I=1, I m=1, is=1, m=1, m me=1, me=1, me is=1                              |
| 13  |        |                    |     | ham            | Da is good go...     | good=2, good good=1, good he=1, he=1, he is=1, is=2, is good=1                  | good=2, good good=1, good he=1, he=1, he is=1, is=2, is...                |
| 14  |        |                    |     | ham            | Dunno dat's w...     | Ok=1, he=1, he me=1, me=1, me Ok=1, s=1, s he=1                                 | Ok=1, he=1, he me=1, me=1, me Ok=1, s=1, s he=1                           |
| 15  |        |                    |     | ham            | Yup ok...            | ok=1  | ok=1  |
| 16  |        |                    |     | ham            | Easy ah'sen go...    | good=1, got=1, got good=1   | good=1, got=1, got good=1   |
| 17  |        |                    |     | spam           | Hey I am really...   | I=1, I am=1, am=1, am want=1, at=1, at to=1, me=1, me text=1, or=...            | I=1, I am=1, am=1, am want=1, at=1, at to=1, me=1, me ...                 |
| 18  |        |                    |     | ham            | V-aluable. A-f...    | I=1, I in=1, in=1   | I=1, I in=1, in=1   |
| 19  |        |                    |     | ham            | Also are you br...   | are=1, are you=1, or=1, you=1, you or=1   | are=1, are you=1, or=1, you=1, you or=1                                   |
| 20  |        |                    |     | ham            | Finally it has h...  | But=1, But t=1, has=1, has is=1, is=1, is now=1, it=1, it has=1, now=...        | But=1, But t=1, has=1, has is=1, is=1, is now=1, it=1, it ha...           |
| 21  |        |                    |     | spam           | Adult 18 Conte...    | be=1, be with=1, will=1, will be=1, with=1, with you=1, you=1                   | be=1, be with=1, will=1, will be=1, with=1, with you=1, y...              |
| 22  |        |                    |     | ham            | Good afternoo...     | How=1, How that=1, is=2, I think=1, I you=1, a=2, a is=1, a from=1, a...        | How=1, How that=1, is=2, I think=1, I you=1, a=2, a is=1, a...            |
| 23  |        |                    |     | ham            | But i dint slept...  | But=1, But is=1, is=1, I in=1, in=1   | But=1, But is=1, is=1, I in=1, in=1                                       |
| 24  |        |                    |     | ham            | R u saying i sh...   | for=1, for it=1, is=2, I the=1, I to=1, it=1, the=1, the is=1, to=1, to for=... | for=1, for it=1, is=2, I the=1, I to=1, it=1, the=1, the is=1, t...       |
| 25  |        |                    |     | ham            | Somewhr som...       | a=1, a time=1, has=1, has a=1, is=1, is u=1, time=1, time to=1, to=1...         | a=1, a time=1, has=1, has a=1, is=1, is u=1, time=1, time ...             |
| 26  |        |                    |     | ham            | Can you call m...    | I=1, I have=1, call=3, call in=1, call me=2, have=2, have call=1, have ...      | I=1, I have=1, call=3, call in=1, call me=2, have=2, have c...            |
| 27  |        |                    |     | ham            | not that i know...   | I=1, I know=1, are=1, are still=1, here=1, here are=1, know=1, know...          | I=1, I know=1, are=1, are still=1, here=1, here are=1, know...            |
| 28  |        |                    |     | ham            | Surely result wi...  | will=1  | will=1  |
| 29  |        |                    |     | ham            | I sent my score...   | I=2, I my=1, I think=1, a=2, a is=1, a on=1, and=1, and is=1, are=1, are...     | I=2, I my=1, I think=1, a=2, a is=1, a on=1, and=1, and is=1...           |
| 30  |        |                    |     | ham            | Hmm ill have t...    | about=1, about it=1, have=1, have to=1, it=1, it ok=1, ok=1, ok you=...         | about=1, about it=1, have=1, have to=1, it=1, it ok=1, ok...              |
| 31  |        |                    |     | ham            | # am i think?        | #=1, # am=1, is=1, I think=1, am=1, am is=1, on=1, think=1, think on=1          | #=1, # am=1, is=1, I think=1, am=1, am is=1, on=1, think...               |
| 32  |        |                    |     | ham            | So how many...       | how=1, how then=1, then=1   | how=1, how then=1, then=1   |
| 33  |        |                    |     | ham            | Didn't you get...    | get=1, get in=1, in=1, t=1, t you=1, you=1, you get=1                           | get=1, get in=1, in=1, t=1, t you=1, you=1, you get=1                     |
| 34  |        |                    |     | ham            | Well the weath...    | But=2, But all=1, But are=1, You=1, You need=1, a=2, a is=1, a to=1...          | But=2, But all=1, But are=1, You=1, You need=1, a=2, a is...              |
| 35  |        |                    |     | ham            | # mins but i h...    | #=1, # but=1, but=1, but is=1, is=1, I to=1, to=1                               | #=1, # but=1, but=1, but is=1, is=1, I to=1, to=1                         |
| 36  |        |                    |     | ham            | Not directly be...   | u=1   | u=1   |
| 37  |        |                    |     | ham            | Your account h...    | #=1, been=1, been by=1, by=1, by is=1, has=1, has been=1, is=2, is ...          | #=1, been=1, been by=1, by=1, by is=1, has=1, has been...                 |
| 38  |        |                    |     | ham            | Omg you can...       | a=1, a in=1, all=1, all the=1, can=1, can a=1, do=1, do get=1, get=1...         | a=1, a in=1, all=1, all the=1, can=1, can a=1, do=1, do g...              |
| 39  |        |                    |     | ham            | Ryder unsold n...    | now=1   | now=1   |
| 40  |        |                    |     | spam           | FreeMsg: Claim...    | for=1, for or=1, now=1, now ur=1, or=1, to=1, to now=1, ur=2, ur fo...          | for=1, for or=1, now=1, now ur=1, or=1, to=1, to now=1, ...               |
| 41  |        |                    |     | ham            | Each Moment L...     | a=1, a u=1, all=1, in=1, in a=1, u=1, u all=1                                   | a=1, a u=1, all=1, in=1, in a=1, u=1, u all=1                             |
| 42  |        |                    |     | ham            | Where did u o...     | oo=1, oo is=1, have=1, have to=1, here=1, in=1, in here=1, is=1, is v...        | oo=1, oo is=1, have=1, have to=1, here=1, in=1, in here=1, is=1, is v...  |

#### 4. 자유 주제: 이미지/텍스트 이진 분류기 (또는 삼진, N-진도 가능)

### 입마개 단속 카메라

#### 1) 아이디어 도출

<뉴스 기사>

**개물림 사고 年 2000건 이상...맹견 입마개 단속 7건 불과**

**반복되는 '개 물림' 사고, 왜... '펫  
티켓' 어디 있나?**

**울산 초등생 '개 물림 사고' 충격...“안 무는 개? 따로 없다!”**

**[개 물림 사고, 근본 대책은] 절대 물지 않는다? 견주들  
과한 자신감부터 버려라**

#### ▶ 문제 정의

: 매년 개 물림 사고가 빈번한데, 자동차의 속도위반 카메라처럼 자동으로 단속하거나 주의 줄 수 없을까?

#### ▶ 해결법 : 입마개 여부를 식별할 수 있는 카메라 설치

- ① 입마개를 한 개와 하지 않은 개를 식별할 수 있도록 모델을 학습시킨다.
- ② 네트워크를 통해 해당 모델과 카메라를 연결한다.
- ③ 입마개를 하지 않은 개를 식별하면 ‘어린이 보호 구역입니다. 반려견에 입마개를 착용시켜주세요.’ 등의 경고 음성을 반복해서 방송한다.
- ④ 해당 구역에서 일정 시간 이상 계속 식별될 시, 관리인에게 알림을 보내 출입을 제한하거나 단속한다.

#### ▶ 설치 장소

: 어린이 보호 구역, 노인 보호 구역, 공공장소 등



사람과 동물을 잇다 애니멀피플

## 2) 이미지 크롤링 -> google colab

<https://colab.research.google.com/drive/1fH12pXaNRPO13YgZLnptzE2uxbF8Bh5M#scrollTo=AL1DdR4P5QNW>

```
# 크롤링한 이미지 train과 test로 분리하여 저장
import os
import shutil
from bing_image_downloader.bing_image_downloader import downloader

directory_list=[
    './custom_dataset/train/',
    './custom_dataset/test/'
]

# 초기 디렉토리 만들기 메소드
def dataset_split(query, train_cnt): # 찾고자 하는 키워드, 개수
    # 학습 및 평가 데이터셋 디렉토리 만들기
    for directory in directory_list:
        if not os.path.isdir(directory+'/'+query):
            os.makedirs(directory+'/'+query)
    # 학습 및 평가 데이터셋 준비
    cnt = 0
    for file_name in os.listdir(query):
        if cnt<train_cnt:
            print(f'[Train Dataset] {file_name}')
            shutil.move(query+'/'+file_name, './custom_dataset/train/'+query+'/'+file_name)
        else:
            print(f'[Test Dataset] {file_name}')
            shutil.move(query+'/'+file_name, './custom_dataset/test/'+query+'/'+file_name)
        cnt = cnt+1
    shutil.rmtree(query)
```

```
[4] # 입마개 하지 않은 강아지 사진 크롤링
query = '강아지'
downloader.download(query, limit=40, output_dir='./', adult_filter_off=True, force_replace=False, #
                    timeout=60)
dataset_split(query, 30)
```

```
[Info] Indexing page: 1
[Info] Indexed 35 Images on Page 1.
=====
#1 이미지 다운로드 (http://cfil206.uf.daum.net/image/2778224F54BA7FA90BD7C8)
#1 파일 다운로드가 완료되었습니다.
#2 이미지 다운로드 (https://img.insight.co.kr/static/2018/11/10/1200/ojyd07r25055j6fen120.jpg)
#2 파일 다운로드가 완료되었습니다.
#3 이미지 다운로드 (https://img4.pann.com/new/download.jsp?FileID=37429723)
#3 파일 다운로드가 완료되었습니다.
```

```
# 입마개 한 강아지 사진 크롤링
query = '개 입마개'
downloader.download(query, limit=40, output_dir='./', adult_filter_off=True, force_replace=False, #
                    timeout=60)
dataset_split(query, 30)
```

```
#35 파일 다운로드가 완료되었습니다.
#36 이미지 다운로드 (http://cfil214.uf.daum.net/image/12594A4B5010CC1C9FE68C)
#36 파일 다운로드가 완료되었습니다.
#37 이미지 다운로드 (https://image-note.pet.akamalized.net/resize/620x-/seImage/20171023/130d816c3eb807749ba8958a28ae3)
#37 파일 다운로드가 완료되었습니다.
#38 이미지 다운로드 (https://img.huffingtonpost.com/asset/5a8cc6fa2000003800eaf768.jpeg?cache=gfxuAMPRB4&ops=cro)
[Error] Invalid image, not saving https://img.huffingtonpost.com/asset/5a8cc6fa2000003800eaf768.jpeg?cache=gfxuAMPRB4&ops=cro
[Info] Issue getting: https://img.huffingtonpost.com/asset/5a8cc6fa2000003800eaf768.jpeg?cache=gfxuAMPRB4&ops=cro
[Error] No active exception to re-raise
```

▶ 다운로드 이미지 확인 후 적절한 이미지 선별

- total : 90
- train images : 30/30
- test images : 10/10

- test
  - 강아지
  - 개 입마개
- train
  - 강아지
  - 개 입마개

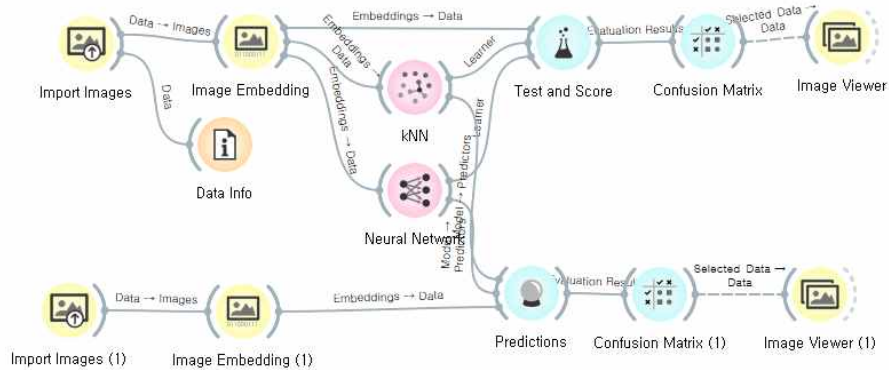


### 3) Orange3 분석

\*\* test 조건

- Image Embedding : Inception v3

- kNN : 7/Euclidean/Uniform // Neural Network : 10, ReLu, L-BFGS-B, 100\_iter



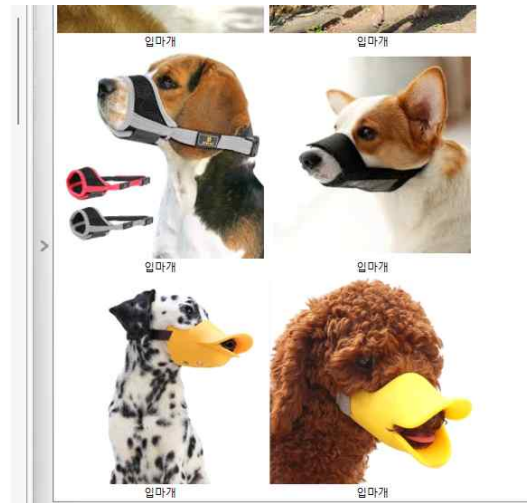
#### ① 1차 test

| Model          | AUC   | CA    | F1    | Precision | Recall |
|----------------|-------|-------|-------|-----------|--------|
| kNN            | 1.000 | 1.000 | 1.000 | 1.000     | 1.000  |
| Neural Network | 1.000 | 1.000 | 1.000 | 1.000     | 1.000  |

| Actual | Predicted |     | Σ  |
|--------|-----------|-----|----|
|        | 강아지       | 입마개 |    |
|        | 강아지       | 입마개 | Σ  |
| 강아지    | 30        | 0   | 30 |
| 입마개    | 0         | 30  | 30 |
| Σ      | 30        | 30  | 60 |

|    | Neural Network     | error | kNN                | error | category |
|----|--------------------|-------|--------------------|-------|----------|
| 1  | 0.00 : 1.00 → 입... | 0.000 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |
| 2  | 0.00 : 1.00 → 입... | 0.000 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |
| 3  | 0.00 : 1.00 → 입... | 0.000 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |
| 4  | 0.00 : 1.00 → 입... | 0.000 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |
| 5  | 0.00 : 1.00 → 입... | 0.000 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |
| 6  | 0.00 : 1.00 → 입... | 0.000 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |
| 7  | 0.00 : 1.00 → 입... | 0.000 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |
| 8  | 0.00 : 1.00 → 입... | 0.000 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |
| 9  | 0.00 : 1.00 → 입... | 0.001 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |
| 10 | 0.08 : 0.92 → 입... | 0.082 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |
| 11 | 1.00 : 0.00 → 강... | 0.004 | 1.00 : 0.00 → 강... | 0.000 | 강아지      |
| 12 | 1.00 : 0.00 → 강... | 0.003 | 1.00 : 0.00 → 강... | 0.000 | 강아지      |
| 13 | 1.00 : 0.00 → 강... | 0.000 | 1.00 : 0.00 → 강... | 0.000 | 강아지      |
| 14 | 1.00 : 0.00 → 강... | 0.000 | 1.00 : 0.00 → 강... | 0.000 | 강아지      |
| 15 | 1.00 : 0.00 → 강... | 0.000 | 1.00 : 0.00 → 강... | 0.000 | 강아지      |
| 16 | 1.00 : 0.00 → 강... | 0.000 | 0.86 : 0.14 → 강... | 0.143 | 강아지      |
| 17 | 1.00 : 0.00 → 강... | 0.000 | 1.00 : 0.00 → 강... | 0.000 | 강아지      |
| 18 | 1.00 : 0.00 → 강... | 0.000 | 1.00 : 0.00 → 강... | 0.000 | 강아지      |
| 19 | 1.00 : 0.00 → 강... | 0.000 | 1.00 : 0.00 → 강... | 0.000 | 강아지      |
| 20 | 1.00 : 0.00 → 강... | 0.000 | 1.00 : 0.00 → 강... | 0.000 | 강아지      |

▶ 너무 좋은 결과에 이상함을 느껴 확인해보니 개의 종이 달랐음. '그래서 결과가 잘 나왔나?' 하는 의심이 들어 맹견 사진으로 2차 테스트를 진행하기로 함. (입마개를 한 개는 거의 맹견)



## ② 2차 test

\*\* 맹견 사진으로 다시 크롤링

```
# 2차 test 맹견 사진 크롤링
query = '맹견'
downloader.download(query, limit=60, output_dir='./', adult_filter_off=True,
                    timeout=60)
dataset_split(query, 450)
```

[Info] Indexing page: 1  
[Info] Indexed 35 Images on Page 1.

Data Info - Orange

Data table properties

Name: untitled

Size: 60 rows, 6 columns

Targets: categorical outcome with 2 classes

Metas: 3 numeric, 2 text

Additional attributes

60



\*\* Test and Score & Confusion Matrix

| Model          | AUC   | CA    | F1    | Precision | Recall |
|----------------|-------|-------|-------|-----------|--------|
| kNN            | 1.000 | 1.000 | 1.000 | 1.000     | 1.000  |
| Neural Network | 1.000 | 1.000 | 1.000 | 1.000     | 1.000  |

|        |     | Predicted |     | Σ  |
|--------|-----|-----------|-----|----|
|        |     | 맹견        | 입마개 |    |
| Actual | 맹견  | 30        | 0   | 30 |
|        | 입마개 | 0         | 30  | 30 |
| Σ      |     | 30        | 30  | 60 |



## \*\* Predictions & Confusion Matrix

| Model          | AUC   | CA    | F1    | Precision | Recall |
|----------------|-------|-------|-------|-----------|--------|
| Neural Network | 0.946 | 0.957 | 0.957 | 0.960     | 0.957  |
| kNN            | 1.000 | 0.957 | 0.957 | 0.960     | 0.957  |

|        |     | Predicted |     |    |
|--------|-----|-----------|-----|----|
|        |     | 맹견        | 입마개 | Σ  |
| Actual | 맹견  | 10        | 0   | 10 |
|        | 입마개 | 1         | 12  | 13 |
| Σ      |     | 11        | 12  | 23 |

### ▶ 오분류 사례 : 입마개 → 맹견

|    | Neural Network     | error | kNN                | error | category |
|----|--------------------|-------|--------------------|-------|----------|
| 13 | 0.02 : 0.98 → 입... | 0.022 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |
| 14 | 1.00 : 0.00 → 맹견   | 1.000 | 0.86 : 0.14 → 맹견   | 0.857 | 입마개      |
| 15 | 0.00 : 1.00 → 입... | 0.000 | 0.00 : 1.00 → 입... | 0.000 | 입마개      |

: 사실 두 번째 테스트에서도 모두 다 맞춰 오분류를 할 때까지 점점 맞추기 어려운 사진들을 test data set에 추가했다. 처음 추가한 것은 입마개와 개의 색이 거의 동일해 구분하기 어려운 사진을 추가했지만, 두 모델 모두 잘 분류했다. 다음으로 추가한 것은 개가 없이 입마개만 있는 사진으로 테스트를 진행했지만, 또 잘 분류했다.

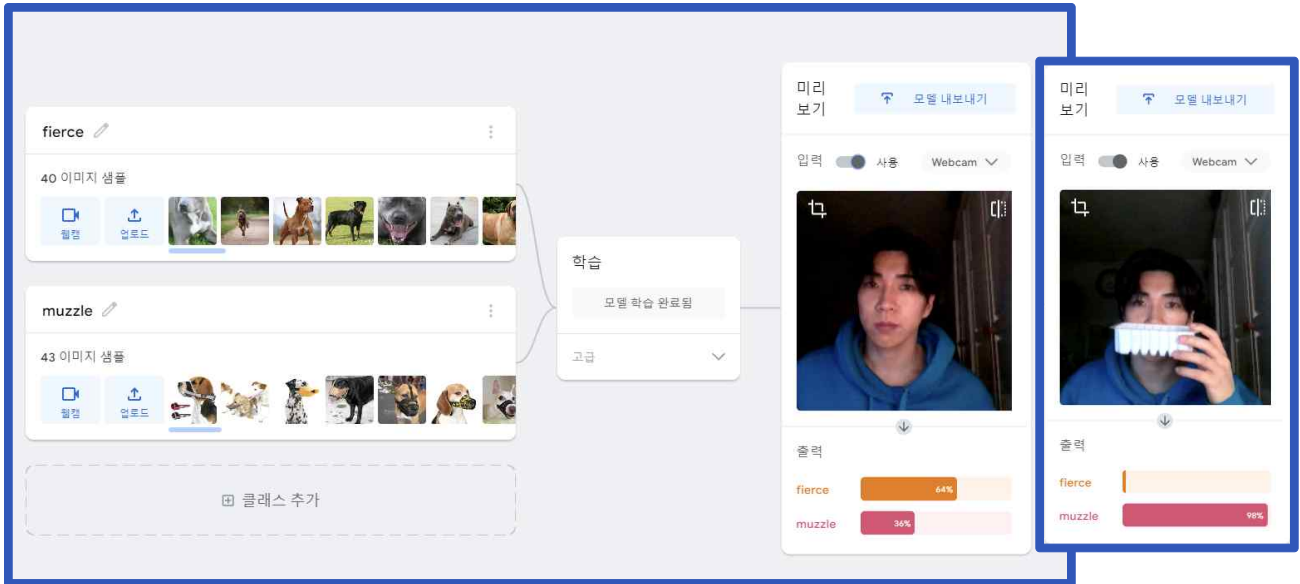
마지막으로 일부로 오분류를 유도하기 위해 입마개가 아주 작게 나온 사진을 추가했고, 그 결과 Neural Network는 1:0로, kNN은 0.86:0.14로 맹견으로 오분류했다.



## 4) Test Web Site

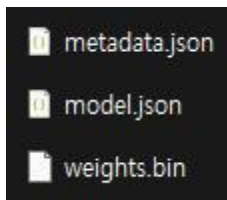
### ① Teachable Machine을 통해 크롤링한 데이터 학습

- fierce : 입마개를 착용하지 않은 개
- muzzle : 입마개를 착용한 개
- ▶ 모델 학습 후, 과자 케이스를 입 앞에 가져다 댔더니 입마개를 한 것으로 분류함



### ② 테스트용 웹 사이트 코딩

- Teachable Machine에서 학습한 위의 모델을 다운로드



- HTML, CSS, JS를 이용해 모델을 사용할 수 있도록 코딩

```
76 // Load the image model and setup the webcam
77 async function init() {
78   const modelURL = URL + "model.json";
79   const metadataURL = URL + "metadata.json";
80
81   model = await tmImage.load(modelURL, metadataURL);
82   maxPredictions = model.getTotalClasses();
83
84   // Convenience function to setup a webcam
85   const flip = true; // whether to flip the webcam
86   webcam = new tmImage.Webcam(500, 500, flip); // width, height, flip
87   await webcam.setup(); // request access to the webcam
88   await webcam.play();
89   window.requestAnimationFrame(loop);
90
91   // append elements to the DOM
92   document.getElementById("webcam-container").appendChild(webcam.canvas);
93   labelContainer = document.getElementById("label-container");
94   for (let i = 0; i < maxPredictions; i++) {
95     // and class labels
96     labelContainer.appendChild(document.createElement("div"));
97   }
98 }
```

```

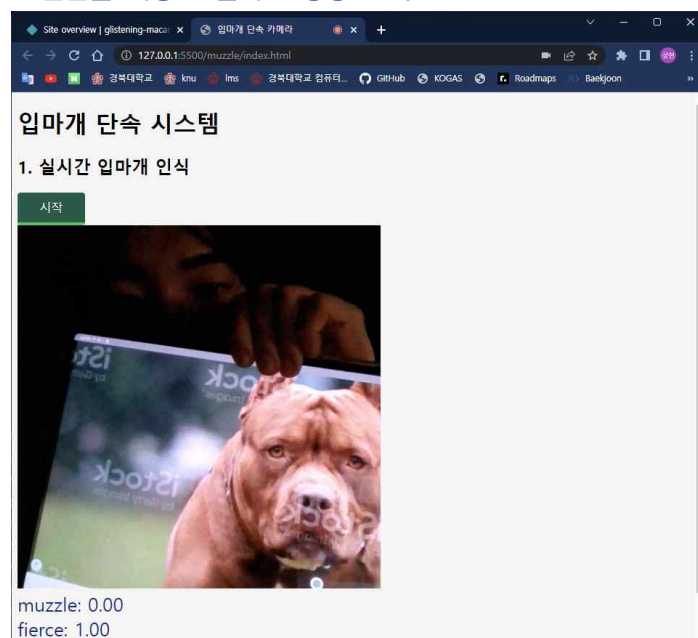
124     async function predict1() {
125         const modelURL = URL + "model.json";
126         const metadataURL = URL + "metadata.json";
127         model = await tmImage.load(modelURL, metadataURL);
128         maxPredictions = model.getTotalClasses();
129         labelContainer1 = document.getElementById("label-container1");
130         for (let i = 0; i < maxPredictions; i++) {
131             // and class labels
132             labelContainer1.appendChild(document.createElement("div"));
133         }
134         // predict can take in an image, video or canvas html element
135         var image = document.getElementById("face-image");
136         const prediction = await model.predict(image, false);
137         var probability1; // 입마개o
138         var probability2; // 입마개x
139
140         for (let i = 0; i < maxPredictions; i++) {
141             if(prediction[i].className == "muzzle") {
142                 probability1 = prediction[i].probability*100
143             } else {
144                 probability2 = prediction[i].probability*100
145             }
146         }
147
148         if (probability1 < probability2){
149             const classPrediction =
150                 "어린이 보호구역입니다. 반려견에 입마개를 착용시켜주세요.\n" +
151                 "(미착용확률: " +
152                 probability2.toFixed(2) +
153                 "%)";
154             //prediction[i].probability.toFixed(2)+"%";
155             labelContainer1.innerHTML = classPrediction;
156         }

```

### ③ Netlify 서버에 입마개 단속 시스템 배포 및 테스트

- 테스트용 주소( 접속 가능 ) : <https://glistening-macaron-c5aa63.netlify.app/>

#### 1. 웹캠을 이용한 실시간 영상 분석



## 2. 사진 업로드 방식 및 경고 문구 출력

### 2. 입마개 사진 업로드

분석하는데 7초 정도 소요됩니다.

예측

ADD IMAGE



REMOVE IMAGE\_4.JPG

어린이 보호구역입니다. 반려견에 입마개를 착용시켜주세요. (미착용확률: 99.92%)

## 4) 후기

이 강의를 듣기 전 인공지능을 활용하기 위해서는 무조건 대학원에 진학해야 한다고 생각했었습니다. 그래서 취업이 목표였던 제게 있어 인공지능은 멀게만 느껴졌었습니다. 하지만 강의를 들으면서 인공지능 모델 자체를 개발하는 것이 아닌, 이미 개발된 인공지능을 활용하는 부분에 있어서는 충분히 가능하겠다는 생각이 들었습니다. 이 생각은 마지막 자유 주제를 진행하며 더 확신이 생겼습니다.

주제를 선정할 때 과제지만 실제로 적용될 수 있는 의미 있는 주제를 고르고 싶었습니다. 인공지능이 우리의 삶에 긍정적인 영향을 주고, 획기적으로 변화시켜주는 곳에 적용하는 것이 옳은 방향이라 생각했기 때문입니다. 그래서 ‘개 물림 사고’ 방지를 개 주인의 양심과 자율에만 맡기는 것이 아니라, 인공지능을 통해 방지해보자는 아이디어에서 입마개 단속 시스템 프로젝트를 생각하게 되었습니다.

크롤링한 데이터로 학습시킨 모델을 분석하고 테스트하는 과정에서 결과가 매우 잘 나오는 것을 확인하고는 학습만 잘 시킨다면 실제로도 활용할 수 있겠다는 생각이 들었습니다. 멀게만 느껴졌던 인공지능을 제가 생각한 아이디어로 구현할 수 있다는 것이 정말 재미있었습니다.

정부에서도 스마트팩토리나 스마트시티 등을 추진하는데 여기에 적용할만한 아이디어만 잘 생각한다면 창업이나 취업에 있어 좋은 기회가 될 것이라는 생각이 들었고, 우리 삶에 여러 문제 중에서 인공지능으로 해결할 수 있는 것들은 어떤 것일지 더 고민해봐야겠습니다.