

MBTI Prediction Based on Text Data

Team 11

한동대학교 유정섭 (ISFP)

숙명여자대학교 이지수 (ENFP)



Introduction

데이터 배경 및 설명



Modeling

데이터 전처리 및 분석



Utilization

모델 활용성

Introduction

Data

MBTI_train.csv

MBTI	TEXT
INTP	say process model list like subscriber channel...
INFJ	upon much manipulate retail finish like sacrific...
...	...

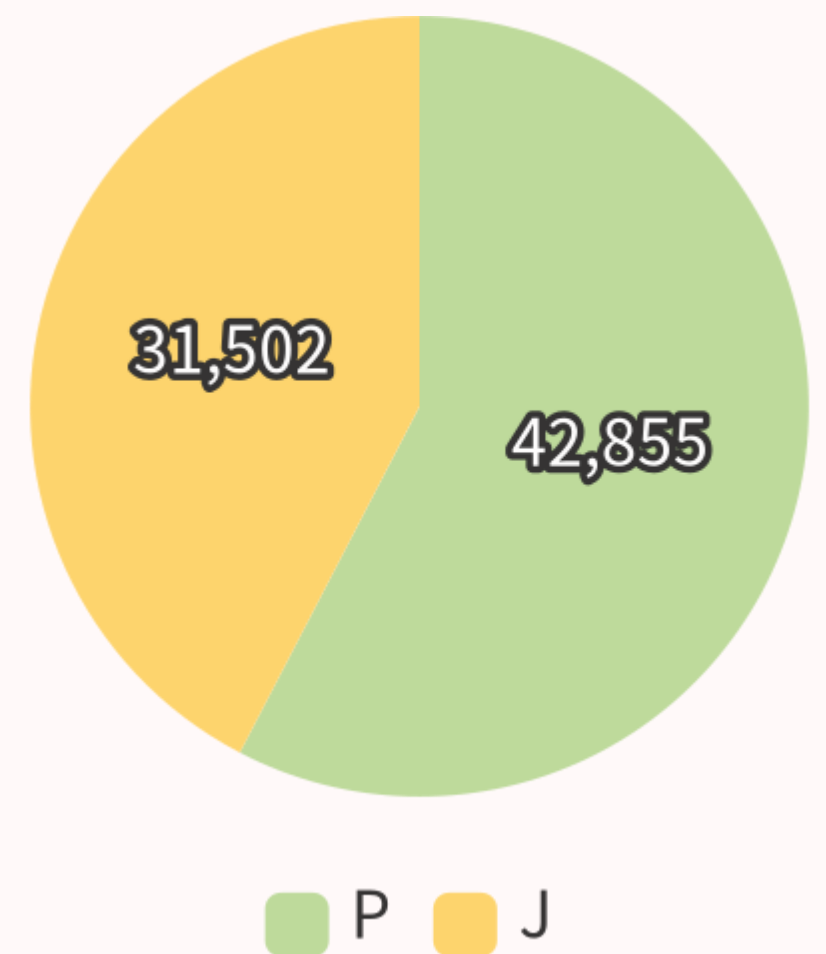
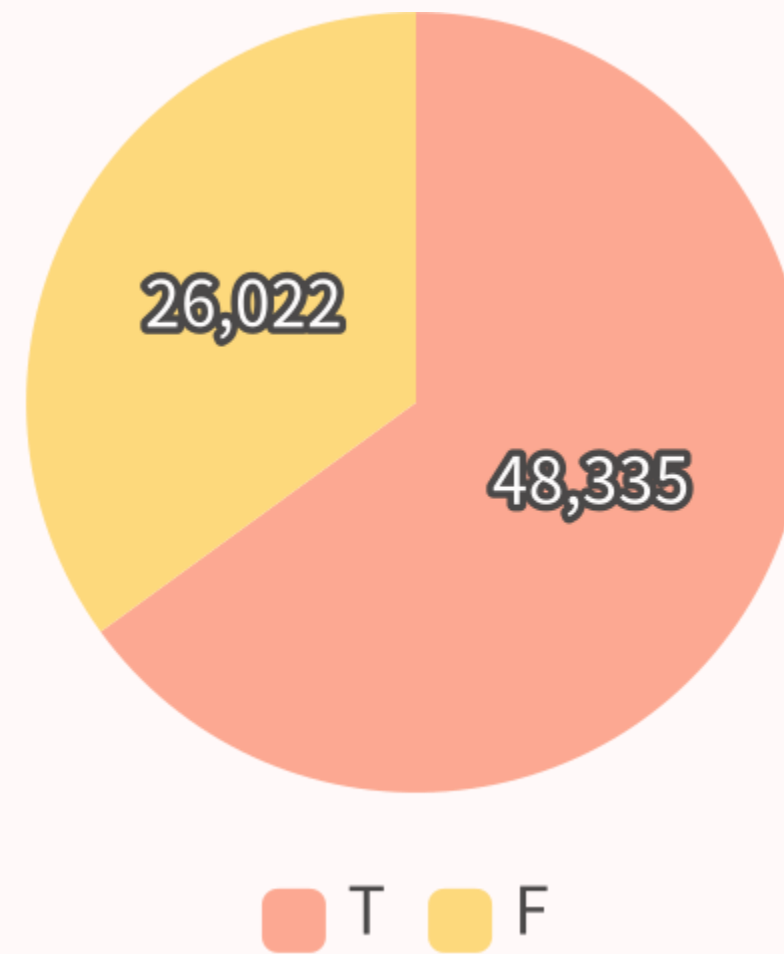
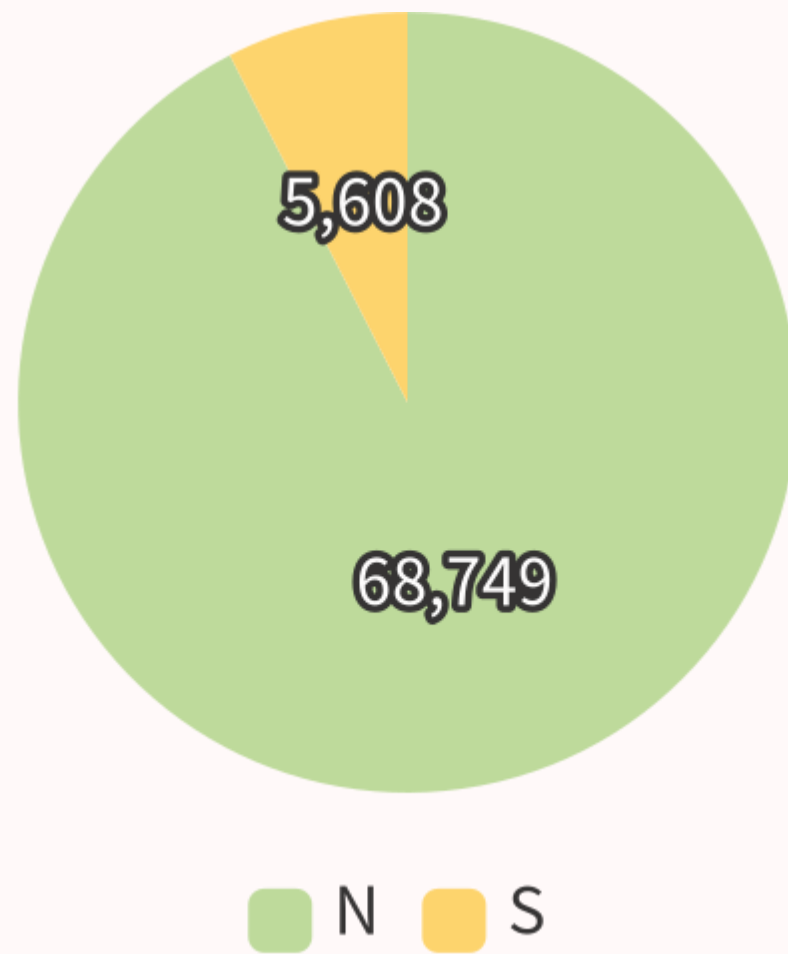
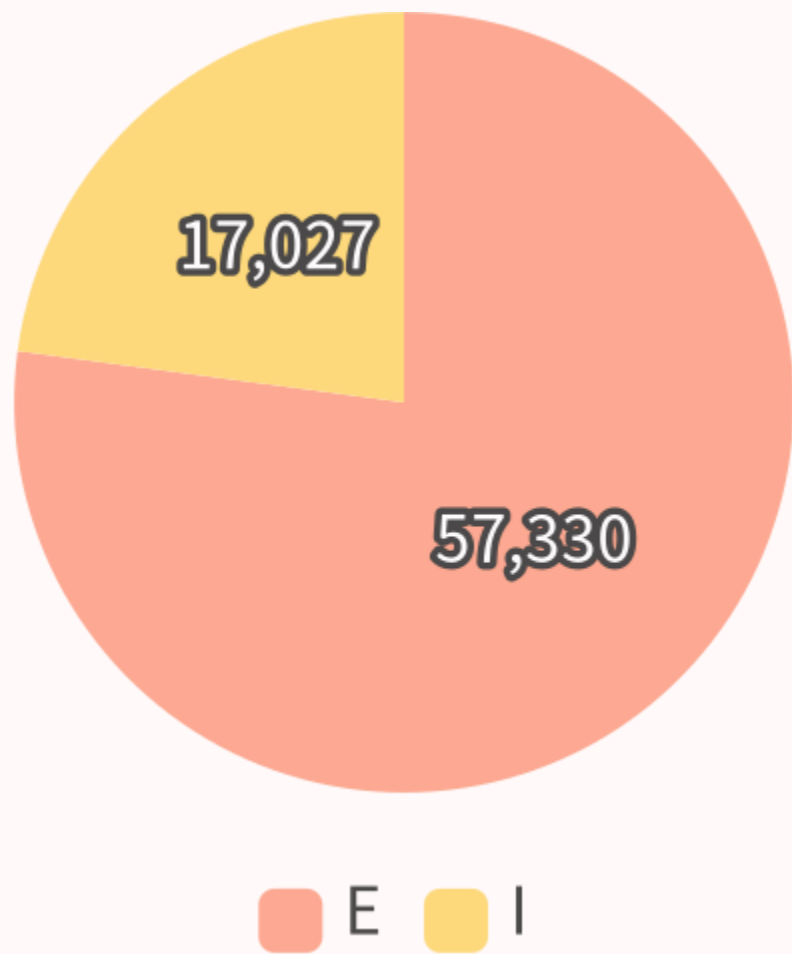
74357 x 2

MBTI_test.csv

TEXT
get accept ya bite well stop important open lo...
offer rebel something war people friend block ...
...

9337 x 1

Data



Modeling

ISFJ

vs

I S F J

Modeling

The letters "ISFJ" are written in a bold, black, serif font. They are enclosed within a hand-drawn, black oval outline that has a slightly irregular, sketchy appearance.

Modeling

LinearSVC GridSearchCV

1. C만 바꿈

```
from sklearn.model_selection import GridSearchCV
```

```
# Training the classifier:  
clf = LinearSVC()  
cv = GridSearchCV(clf, {'C': [0.1, 0.5, 1.0]})  
  
text_clf = Pipeline([('tfidf', TfidfVectorizer()), ('clf', cv)])  
text_clf.fit(X_train, y_train)  
  
C = cv.best_estimator_.C
```

```
cv.best_estimator_.C
```

```
0.5
```

나이브베이즈 분류기

- MultinomialNB

```
from sklearn.naive_bayes import MultinomialNB
```

```
clf = MultinomialNB().fit(X_train_tfidf, y_train)  
  
text_clf = Pipeline([('tfidf', TfidfVectorizer()), ('clf', MultinomialNB())])  
text_clf.fit(X_train, y_train)  
  
Pipeline(steps=[('tfidf', TfidfVectorizer()), ('clf', MultinomialNB())])  
  
predictions = text_clf.predict(X_test)
```


Modeling

로지스틱회귀 분류기

```
from sklearn.linear_model import LogisticRegression
```

```
clf = LogisticRegression(random_state=0, max_iter=10000)
clf.fit(X_train_tfidf, y_train)
```

```
text_clf = Pipeline([('tfidf', TfidfVectorizer()), ('clf', LogisticRegression(random_state=0, max_iter=10000))])
text_clf.fit(X_train, y_train)
```

```
Pipeline(steps=[('tfidf', TfidfVectorizer()), ('clf', LogisticRegression(max_iter=10000, random_state=0))])
```

```
predictions = text_clf.predict(X_test)
accuracy_score(predictions, y_test)
```

0.7829478214093599

랜덤포레스트 분류기

```
from sklearn.ensemble import RandomForestClassifier
```

```
clf = RandomForestClassifier(n_estimators=200, max_depth=6, random_state=0)
clf.fit(X_train_tfidf, y_train)
```

Pipelining the vectorizer and the classifier

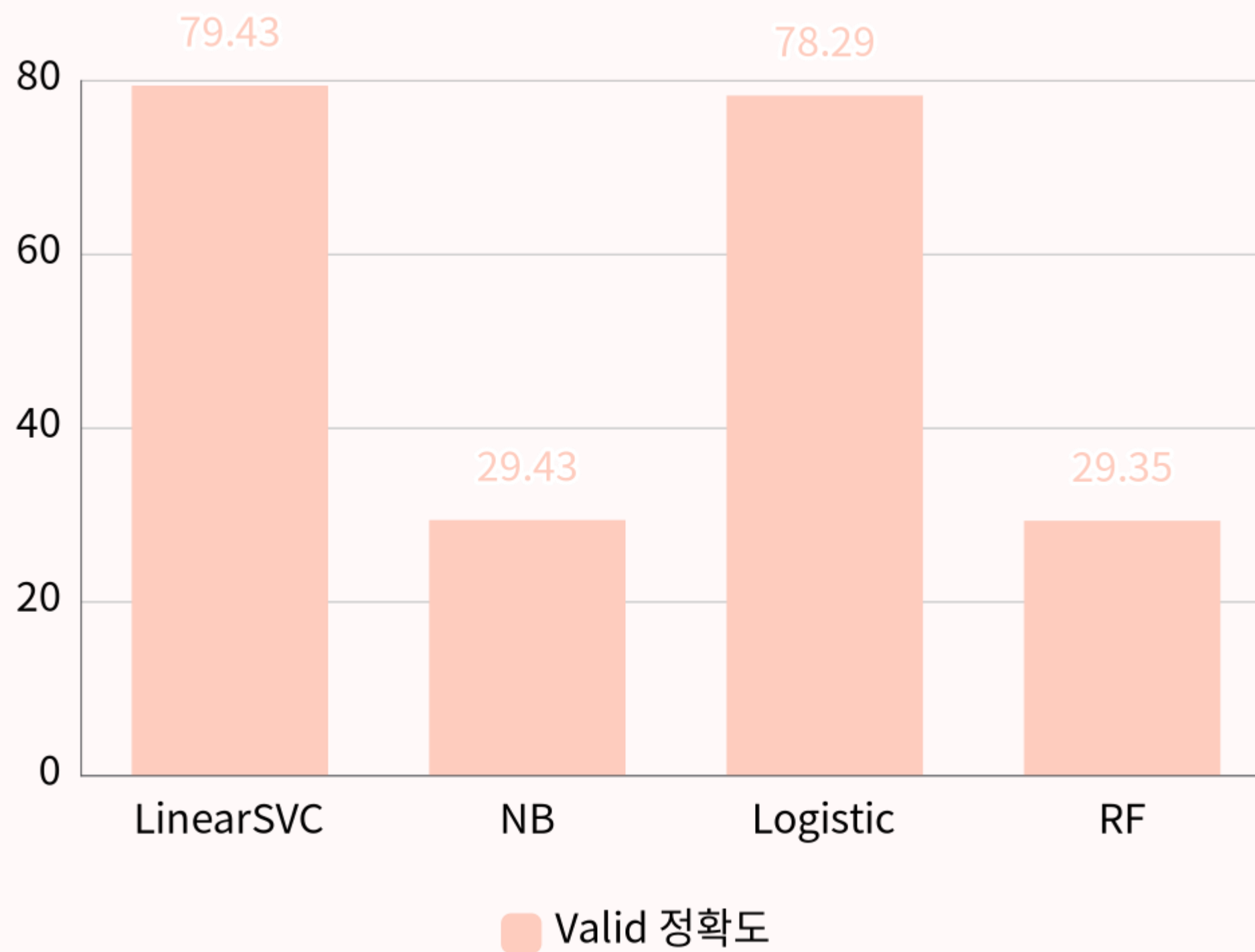
```
text_clf = Pipeline([('tfidf', TfidfVectorizer()), ('clf', RandomForestClassifier(n_estimators=200, max_depth=6, random_state=0))])
text_clf.fit(X_train, y_train)
```

```
Pipeline(steps=[('tfidf', TfidfVectorizer()), ('clf', RandomForestClassifier(max_depth=6, n_estimators=200, random_state=0))])
```

```
predictions = text_clf.predict(X_test)
accuracy_score(predictions, y_test)
```

0.2935045723507262

Modeling



Modeling

LinearSVC GridSearchCV

1. C만 바꿈

```
from sklearn.model_selection import GridSearchCV
```

```
# Training the classifier:
clf = LinearSVC()
cv = GridSearchCV(clf, {'C': [0.1, 0.5, 1.0]})

text_clf = Pipeline([('tfidf', TfidfVectorizer()), ('clf', cv)])
text_clf.fit(X_train, y_train)

C = cv.best_estimator_.C
```

```
cv.best_estimator_.C
```

```
0.5
```

로지스틱회귀 분류기

```
from sklearn.linear_model import LogisticRegression
```

```
clf = LogisticRegression(random_state=0, max_iter=10000)
clf.fit(X_train_tfidf, y_train)
```

```
text_clf = Pipeline([('tfidf', TfidfVectorizer()), ('clf', LogisticRegression(random_state=0, max_iter=10000))])
text_clf.fit(X_train, y_train)
```

```
Pipeline(steps=[('tfidf', TfidfVectorizer()),
                  ('clf', LogisticRegression(max_iter=10000, random_state=0))])
```

```
predictions = text_clf.predict(X_test)
accuracy_score(predictions, y_test)
```

Modeling

MBTI	TEXT
INTP	say process model list like subscriber channel...
INFJ	upon much manipulate retail finish like sacrific...
...	...



**Tf-Idf
Vectorizer**

Model_1

```
# 모든 설명변수 데이터 X 자연어처리  
X_tfidf = tfidf.fit_transform(X)
```

```
svc_clf = Pipeline([('tfidf', TfidfVectorizer()), ('clf', LinearSVC(C=0.3))])  
svc_clf.fit(X, y)
```

```
Pipeline(steps=[('tfidf', TfidfVectorizer()), ('clf', LinearSVC(C=0.3))])
```

```
predictions_svc = svc_clf.predict(test['text'])
```

```
test_pred = pd.DataFrame(predictions_svc)
```

TfidfVectorizer()
LinearSVC(C=0.3)

Model_2

```
# StopWord 확인
nlp = spacy.load('en_core_web_sm')
print("불러온 Stopword (무의미한 단어)", len(nlp.Defaults.stop_words))
def remove_stopwords(s):
    new_s = []
    for word in s.split():
        if not nlp.vocab[word].is_stop:
            new_s.append(word)
    return ' '.join(new_s)
```

불러온 Stopword (무의미한 단어) 326

```
# Lemmatization(표제어 추출 : 단어로부터 표제어 찾기)

s_stemmer = SnowballStemmer(language='english')
def replace_stemwords(s):
    new_s = []
    for word in s.split():
        new_s.append(s_stemmer.stem(word))
    return ' '.join(new_s)
```

무의미한 단어 제거 및 표제어 추출



```
# 모든 설명변수 데이터 X 자연어처리
X_tfidf_second = tfidf.fit_transform(X_second)

svc_clf = Pipeline([('tfidf', TfidfVectorizer()), ('clf', LinearSVC(C=0.3))])
svc_clf.fit(X_second, y_second)

Pipeline(steps=[('tfidf', TfidfVectorizer()), ('clf', LinearSVC(C=0.3))])

predictions_svc_2 = svc_clf.predict(test.text_second)

test_pred_2 = pd.DataFrame(predictions_svc_2)
```

TfidfVectorizer()
LinearSVC(C=0.3)

Model_1 + Model_2

	A	B	C	D
1	ENFP			
2	ENTP			
3	INTJ			
4	INTJ			
5	INTJ			
6	INTJ			
7	INFP			
8	INFJ			
9	ENFP			
10	INFJ			
11	ENTP			

Utilization




1. 콜센터

1. 콜센터



1. 콜센터



초보운전 세삼


헉

나 차 사고 났어ㅠㅠㅠㅠㅠㅠ
어떡해ㅠㅠㅠㅠ 무서워ㅠㅠ

너 보험 들었지?
보험사에 얼른 전화해

사고 났을 때는 보험사 직원
빨리 부르는 게 중요함

거기 어디쯤인지 파악해서 보험사 전화 ㄱㄱ



초보운전 세삼

헉

나 차 사고 났어ㅠㅠㅠㅠㅠㅠ
어떡해ㅠㅠㅠㅠ 무서워ㅠㅠ

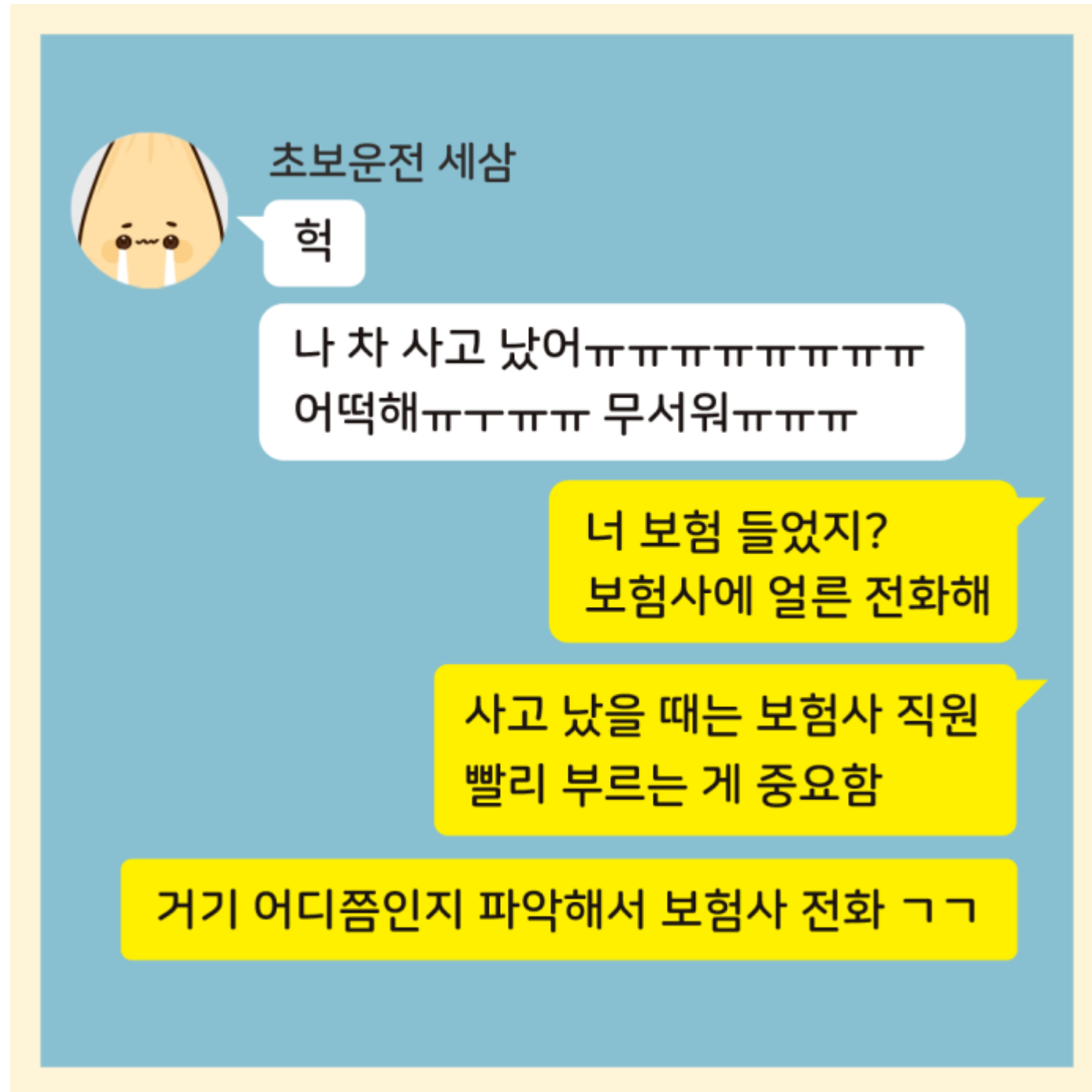
헉ㅠㅠ 너 괜찮아??

다치진 않았어??

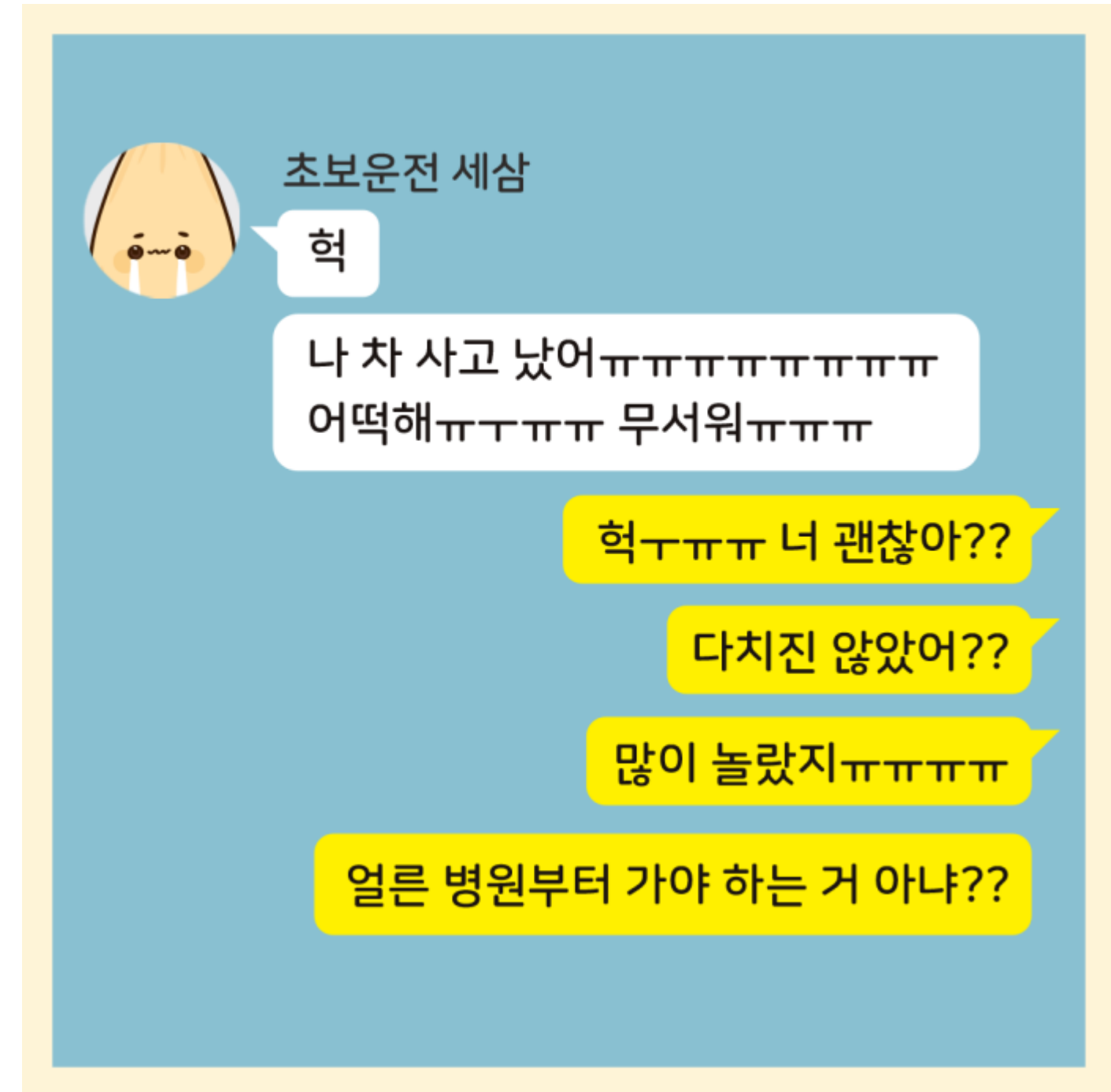
많이 놀랐지ㅠㅠ

얼른 병원부터 가야 하는 거 아냐??

1. 콜센터



사고형 (T)



감정형 (F)



2. 마케팅

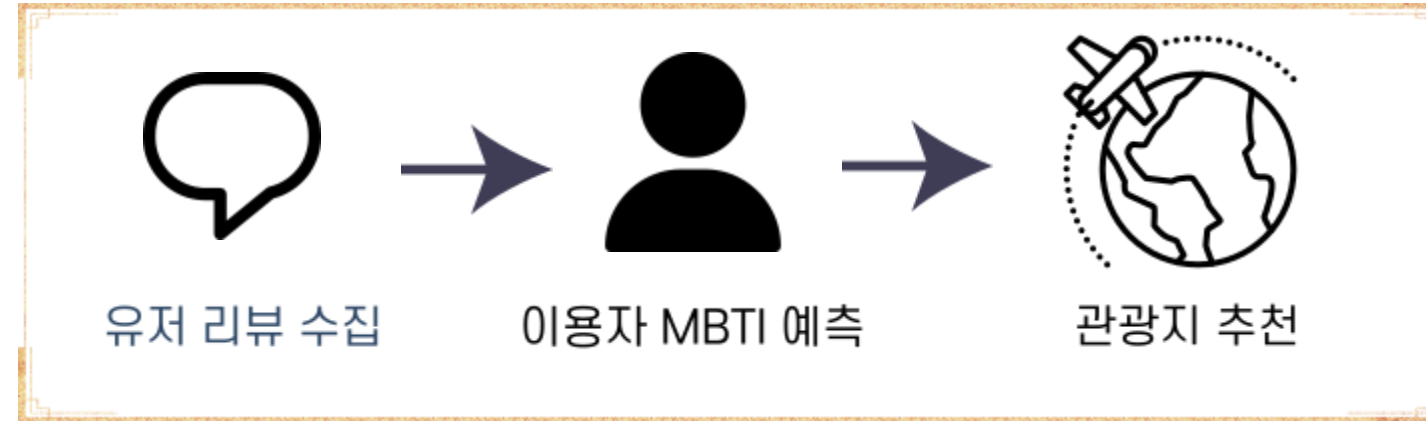
2. 마케팅



첫째, 개인의 성격 유형은 광고에 대한 태도를 평가함에 있어 영향을 미치는 요인으로 작용하는 것으로 나타났다. 또 이들은 광고 유형에 따라 각기 다르게 반응하는 것으로 나타났다. 즉, 이미지 광고는 SF유형이, 외향적-기능성 광고는 NT유형의 성격이 광고에 대한 태도에 보다 긍정적인 반응을 보이는 것으로 나타났다. 이러한 결과는 개인 소비자의 성격 유형과 광고의 이미지가 일치할수록 광고에 대한 태도는 보다 긍정적이 된다는 시사점을 제시하고 있는데, 이를 통하여 광고전략 수립 시 타겟 프로필에 성격 유형을 반영할 수 있다는 점과, 성격 유형이 시장의 세분화 요인으로 작용하는 것은 보다 효율적인 광고 전략 수립에 일조할 수 있을 것으로 보인다.



2. 마케팅



연구 결과는 첫째, 관광동기 요인 중 일상탈출과 휴식은 성격 별로 유의한 차이를 보이지 않았다. 두 가지 요인은 관광을 하는 것에 가장 궁극적인 목적이기 때문에 차이를 보이지 않는 것으로 사료된다. 둘째, 호기심 요인은 합리자형(-NT)이 유의미한 결과를 나타냈다. 합리자형(-NT)은 호기심 요인이 가장 높으며 무언가에 대한 욕구가 높고 새로운 것이나 기존 틀에 벗어나는 것에 대한 잠재적 욕구가 있는 것을 알 수 있다. 셋째, 가족화합 요인은 보호자형(-SJ)이 유의미한 결과를 나타냈다. 보호자형(-SJ)은 가족화합 요인이 가장 높으며 가족과 화합하고 인간관계 및 소속감을 중요시 한다는 것으로 해석할 수 있다. 넷째, 경험성 요인은 이상가형(-NF)이 유의미한 결과를 나타냈다. 이상가형(-NF)은 경험성 요인이 제일 높으며 무언가를 경험하려는 욕구가 크다는 것으로 해석할 수 있다. 다섯째, 유희성은 예술가형(-SP)이 유의미한 결과를 나타냈다. 예술가형(-SP)은 유희성 요인이 제일 높으며 자유분방하며 즐거움을 추구하는 것으로 해석할 수 있다. 이 결과로 보아 합리자형(-NT), 이상가형(-NF), 보호자형(-SJ), 예술가형(-SP)은 모두 관광 동기 부분에서 유의미한 결과를 나타냈으며 이에 따른 관광지 추천이 가능한 것을 시사해 주고 있다.



Conclusion

Model Demonstration