






# Week4

 Created By	 DongGu Kim
 Last Edited	@Apr 17, 2020 7:21 PM
 Property	
 Tags	

1. Mecab 설치
2. Directory 변경
3. 형태소 분류 칼럼 추가 (Mecab 이용)
4. EDA(탐색적 데이터 분석)
5. TF-IDF 분석
6. 스미싱·일반 문자에서 많이 쓰인 단어 Top100 비교

---

## 1. Mecab 설치

```
!pip install git

! git clone https://github.com/SOMJANG/Mecab-ko-for-Google-Colab.git

cd Mecab-ko-for-Google-Colab/

ls

! bash install_mecab-ko_on_colab190912.sh
```

---

## 2. Directory 변경

```
from google.colab import drive

drive.mount('/content/gdrive')

from konlpy.tag import Mecab
mecab = Mecab()
from tqdm import tqdm_notebook

ls
```

```
cd ../gdrive/My\ Drive/kb_data
```

### 3. 형태소 분류 칼럼 추가(Mecab 이용)

```
train['morph'] = 0

%%time
for idx in tqdm_notebook(range(len(train))):
    train['morph'][idx] = mecab.morphs(train['text'][idx])

# to_csv 지정
#train.to_csv("morph.csv")
```

### 4. EDA (탐색적 데이터 분석)

```
train['smishing'].value_counts()

pos_train = train[train['smishing']==1]
neg_train = train[train['smishing']==0]

print("스미싱 문자 비율: ", len(pos_train)/len(train)*100, '%')
print(" 일반 문자 비율: ", len(neg_train)/len(train)*100, '%')
```

- Labeling Data 비율
  - 스미싱 문자 6.5% (1.8만 건)
  - 일반 문자 93.5% (27만 건)

### Labeling Data 비율

- 스미싱 문자 6.5% (1.8만 건)
- 일반 문자 93.5% (27만 건)

```
[ ] train['smishing'].value_counts()
```

```
0    277242  
1     18703  
Name: smishing, dtype: int64
```

```
[ ] pos_train = train[train['smishing']==1]  
    neg_train = train[train['smishing']==0]
```

```
[ ] print("스미싱 문자 비율: ", len(pos_train)/len(train)*100, '%')  
    print(" 일반 문자 비율: ", len(neg_train)/len(train)*100, '%')
```

```
스미싱 문자 비율:  6.31975535994864 %  
일반 문자 비율:  93.68024464005136 %
```

### 문자 메시지 길이 평균

- 스미싱 문자: 801
- 일반 문자: 133

→ 스미싱 문자가 일반 문자에 비해 매우 긴 것으로 드러남.

```
pos_text_sum = 0  
for i in pos_train['text']:  
    pos_text_sum += len(i)  
print(pos_text_sum/len(pos_train['text']))
```

```
neg_text_sum = 0  
for i in neg_train['text']:  
    neg_text_sum += len(i)  
print(neg_text_sum/len(neg_train['text']))
```

## 5. TF-IDF 분석

```
# DTM 문서 단어 행렬
from sklearn.feature_extraction.text import CountVectorizer
vector = CountVectorizer()

# TF-IDF
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.feature_extraction.text import TfidfTransformer
tfvector = TfidfTransformer(smooth_idf=False)
```

## 1) DTM (문서 단어 행렬, Document-Term Matrix, DTM)

```
tot_to_tf = vector.fit_transform(train2['morph'])
tot_vocab = vector.get_feature_names()
tot_bow = pd.DataFrame(tot_to_tf.toarray(), columns=tot_vocab).head()
dist = np.sum(tot_to_tf, axis=0)
tot_freq = pd.DataFrame(dist, columns=tot_vocab)
tot_top30 = tot_freq.T.sort_values(by=0, ascending=False).head(30)
tot_top30.to_csv("전체_탑30개 단어(bow).csv", encoding='utf-8')
```

xxx	86210
은행	16390
고객	14256
올림	12539
세요	12033
리브	8392
지점	8357
슬니다	8248
합니다	8244
으로	8119
드립니다	7626

## 2) TF-IDF (Term Frequency-Inverser Document Frequency)

### ㄱ. 스미싱 문자 TF-IDF 분석

```
pos_to_tf = vector.fit_transform(pos_train['morph'])
pos_to_tf.shape
```

```

pos_vocab = vector.get_feature_names()
print(len(pos_vocab))

pos_bow = pd.DataFrame(pos_to_tf.toarray(), columns=pos_vocab).head()

dist = np.sum(pos_to_tf, axis=0)

pos_freq = pd.DataFrame(dist, columns=pos_vocab)
pos_freq

pos_top100 = pos_freq.T.sort_values(by=0, ascending=False).head(100)
pos_top100.to_csv("스미싱_탑100개 단어(bow).csv", encoding='utf-8')

%%time
pos_tfidf = tfvector.fit_transform(pos_to_tf)
pos_tfidf.shape

# 각 row에서 전체 단어가방에 있는 어휘에서 등장하는 단어에 대한 one-hot-vector에 TF-IDF 가중치 반영
pos_tfidf_freq = pd.DataFrame(pos_tfidf.toarray(), columns=pos_vocab)
pos_tfidf_freq.head()

pos_tfidf = pd.DataFrame(pos_tfidf_freq.sum())
pos_tfidf_top = pos_tfidf.sort_values(by=0, ascending=False)
pos_tfidf_top.head(30)

pos_tfidf_top100 = pos_tfidf_top.head(100)
pos_tfidf_top100.to_csv("스미싱_탑100개 단어(tfidf).csv", encoding='utf-8')

```

- 스미싱 문자에서 자주 쓰인 단어
  - (TF-IDF로 공통적으로 자주 등장하는 단어는 배제함)

xxx	3122.235633
가능	1842.781441
상품	1823.406129
으로	1788.195767
금리	1673.120734
대출	1657.600631
등급	1633.971755
상담	1433.496556
습니다	1405.077144
한도	1241.518391
은행	1176.026641
까지	1168.540788
신용	1167.314933
드립니다	1154.395195
신청	1086.859317

## ㄴ. 일반 문자 TF-IDF 분석

```

neg_to_tf = vector.fit_transform(neg_train['morph'][:18702])
neg_to_tf.shape

neg_vocab = vector.get_feature_names()
print(len(neg_vocab))

neg_bow = pd.DataFrame(neg_to_tf.toarray(), columns=neg_vocab).head()

dist = np.sum(neg_to_tf, axis=0)

neg_freq = pd.DataFrame(dist, columns=neg_vocab)
neg_freq

neg_top100 = neg_freq.T.sort_values(by=0, ascending=False).head(100)
neg_top100.to_csv("일반_탑100개 단어(bow).csv", encoding='utf-8')

%%time
neg_tfidf = tfvector.fit_transform(neg_to_tf)
neg_tfidf.shape

# 각 row에서 전체 단어가방에 있는 어휘에서 등장하는 단어에 대한 one-hot-vector에 TF-IDF 가중치 반영
neg_tfidf_freq = pd.DataFrame(neg_tfidf.toarray(), columns=neg_vocab)
neg_tfidf_freq.head()

```

```
neg_tfidf = pd.DataFrame(neg_tfidf_freq.sum())
neg_tfidf_top = neg_tfidf.sort_values(by=0, ascending=False)
neg_tfidf_top.head(30)
```

일반 문자에서 자주 쓰인 단어

- (TF-IDF로 공통적으로 자주 등장하는 단어는 배제함)

xxx	3094.942393
세요	901.017975
은행	886.626054
고객	809.616712
올림	786.176764
행복	699.462992
합니다	653.342119
감사	606.765739
습니다	604.888097
리브	599.388555
지점	589.791605
드립니다	587.462612
으로	577.292284
입니다	544.479502
보내	540.858305

## 6. 스미싱·일반 문자에서 많이 쓰이는 Top100 비교

```
pos_wd = list(dict(pos_tfidf_top100[0]).keys())
len(pos_wd)

neg_wd = list(dict(neg_tfidf_top100[0]).keys())
len(neg_wd)

same_list = list()
diff_pos_list = list()
diff_neg_list = list()
```

```

for pos in pos_wd:
    if pos in neg_wd:
        same_list.append(pos)
    else:
        diff_pos_list.append(pos)

for neg in neg_wd:
    if neg not in pos_wd:
        diff_neg_list.append(neg)

```

```

print(len(same_list), same_list)
print()
print('스미싱', len(diff_pos_list), diff_pos_list)
print()
print('일 반', len(diff_neg_list), diff_neg_list)

```

- 31개 단어가 공통적으로 사용되었으며, 나머지 69개 단어는 달랐음.

```

31 ['xxx', '가능', '으로', '대출', '습니다', '은행', '까지', '드립니다', '합니다', '드리', '거부', '이용', '수신', '금융',
스미싱 69 ['상품', '금리', '등급', '상당', '한도', '신용', '신청', '진행', '부채', '통합', '이상', '대환', '문자', '추거
일 반 69 ['올림', '행복', '리브', '지점', '주말', '가득', '팀장', '직원', '바랍니다', '건강', '하루', '스토어', '포인트

```