


Week9

Created By	 DongGu Kim
Last Edited	@May 15, 2020 7:27 PM
Property	
Tags	

Using Lime for Interpretable Machine Learning

Django를 활용한 Smishing 감지 웹사이트 제작

Using Lime for Interpretable Machine Learning

1. 필요한 라이브러리 설치 및 호출

```
#Install non-standard packages (assuming jupyter notebook)
# 필요한 라이브러리 설치
!pip install shap
!pip install lime
!pip install eli5

from lime import lime_text
from lime.lime_text import LimeTextExplainer
from sklearn.pipeline import make_pipeline
```

2. 모델 장착

```
X_origin = vec_train
Y_label = prepared_train['smishing']
lgbm_model = joblib.load('lgbm_models/[4125, 2866, 3885]/'+lgbm_model)
```

3. LIME 이용

```
class_names = ['normal', 'smishing']
explainer = LimeTextExplainer(class_names=class_names)
classifier = make_pipeline(vectorizer, lgbm_model)
```

```
idx = 999
explanation = explainer.explain_instance(prepared_train['origin'][idx], classifier.predict_proba, num_features=6)
print('Document id: %d' % idx)
print('Probability(smishing) =', classifier.predict_proba([prepared_train['origin'][idx]])[0,1])
```

```
/usr/local/lib/python3.6/dist-packages/lime/lime_text.py:114: FutureWarning: split() requires a non-empty pattern match.
  self.as_list = [s for s in splitter.split(self.raw) if s]
Document id: 999
Probability(smishing) = 4.8679970823781045e-06
```

```
%matplotlib inline

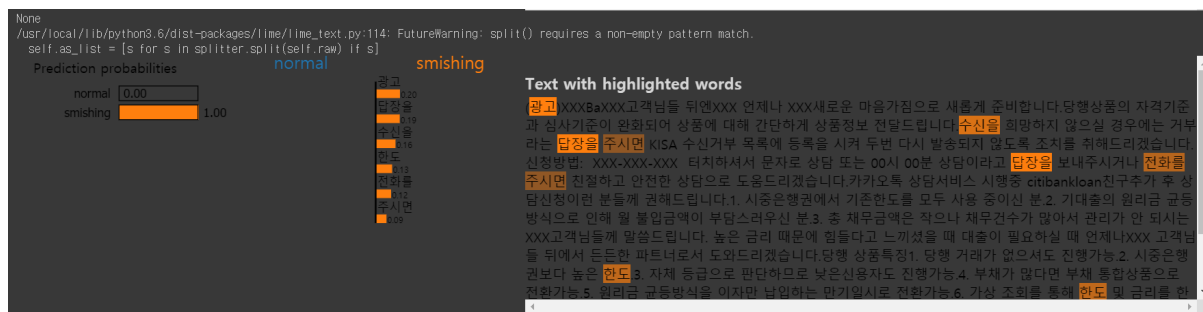
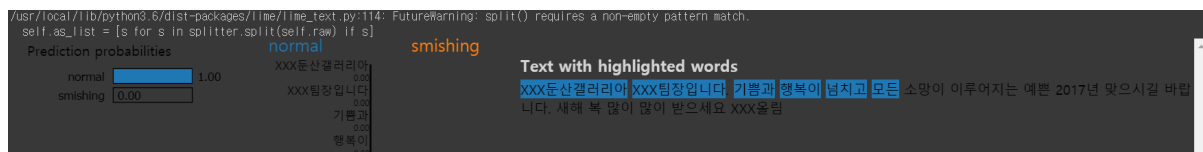
for idx in range(10):
    explanation = explainer.explain_instance(prepared_train['origin'][idx], classifier.predict_proba, num_features=6)
    print(explanation.show_in_notebook(text=True))
```

4. smishing 여부를 가려내는데 유효한 키워드

- Input(단어)를 일부 빼고 Output(예측값)이 변하는 정도로, 중요성을 판단
- 참고

<https://dreamgonfly.github.io/2017/11/05/LIME.html>

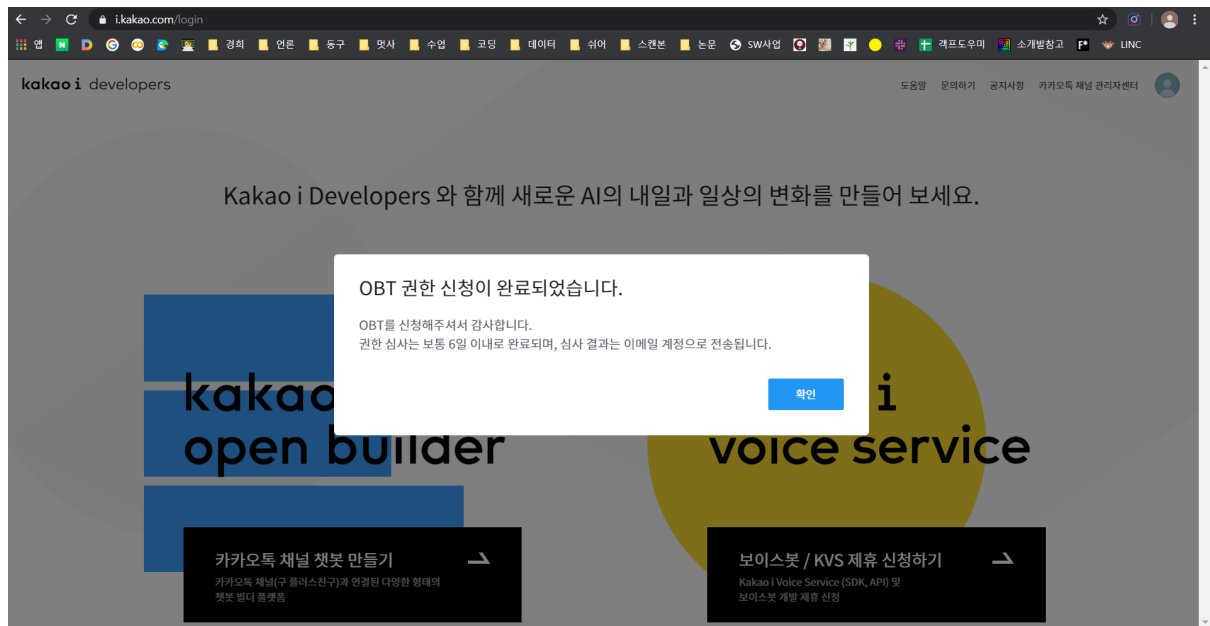
<https://dreamgonfly.github.io/2017/11/05/LIME.html>



2. Django를 활용한 Smishing 감지 웹사이트 제작

방법1 : 카카오톡 오픈빌더 (Django)

- 플러스친구 계정 생성 및 오픈베타 신청



- 오픈API를 이용한 챗봇은 2019년 12월부로 서비스 종료

방법2: 라인 챗봇 (line-bot-sdk python)

- line-bot-sdk python는 flask 프레임워크에 최적화됨
- django로도 구현 가능하지만 url, callback을 별도로 수정해줘야 함.

1) LINE 챗봇 서비스 구축: https://blog.naver.com/n_cloudplatform/221245743135

2) LINE 비즈니스 계정 생성

LINE@ 계정을 신청해 주셔서 감사합 니다.

아래 정보를 바탕으로 계정을 만들었습니다.
LINE@ 계정을 설정하려면 LINE 계정 정보를 입력하여 로그인한 다음 관리화면으
로 이동해 주세요.

LINE@ Manager에 로그인

3) 메시지 API Docs

<https://developers.line.biz/en/reference/messaging-api/>
<https://developers.line.biz/en/reference/messaging-api/>

4) line-bot sdk python github

<https://github.com/line/line-bot-sdk-python/blob/master/examples/flask-kitchensink/app.py>
<https://github.com/line/line-bot-sdk-python/blob/master/examples/flask-kitchensink/app.py>

- 결론

- Flask는 이용 경험이 없어서 Django를 사용할 예정
- Django로 카카오톡, LINE 모두 이용 가능하기 때문에 우선 Django로 웹서비스를 구현할 예정.