[1. Introduction : Trusting]

이 논문에서는 결과를 신뢰하는 것을 두가지로 나누어서 정의한다.

[1.1 Trusting model]

1.1.1 Trusting prediction

개별 예측을 믿고 이를 바탕으로 행동을 취하는 것을 말한다.
개별 예측에 대한 신뢰는 예측값에 기반한 의사결정 문제에서 매우 중요한 요소이다.
ex) 예컨대 의료 진단을 하거나 테러범을 탐지하는 문제에서 예측 과정에 대한 이해 없이 그저 모델이 예측한 값을 토대로 판단하는 것은 매우 위험할 것. 진단을 내리는데 결정적인 역할을 한 증상이 무엇인지, 어떤 특징이 테러리스트라고 판단하는데 영향을 주었는지를 알 수있다면 사용자가 결과를 신뢰할지 신뢰하지 않을지 판단하는데 도움이 될 수 있다.

1.1.2. Trusting a model

모형 전반에 대한 신뢰를 말한다.
즉 이 모델이 전체적으로 믿을만 한지에 대한 것이다.
모델이 우리가 가진 데이터 외에도, 실제 데이터를 적용하였을 때에도 잘 작동하는지 알고싶다.
머신러닝 모델을 만들때 이를 위해서 일반적으로 validation 또는 test set에 대한 Accuracy Metric을 이용해서 accuracy를 측정한 뒤, 이 값을 토대로 모델의 성능을 추정한다.
이 방법은 유용하긴 하지만 실제 시스템에 적용하여 실데이터의 예측값을 산출했을 때 보다 정확도를 과대 측정하는 경우가 많다. (에러가 많아)

※ 참고
아래 항목들은 잘못된 모델 평가를 일으키는 대표적인 원인들이다.
Data leakage : 의도하지 않은 잘못된 시그널이 훈련/검증 데이터 셋에 포함되는 경우. 예를 들어 환자의 id와 발병여부는 매우 높은 상관관계를 가지지만 의미가 없는 잘못된 데이터임. 이것이 실수로 데이터셋에 포함된다면 모델은 의미 없는 패턴을 학습하게 됨.
Data shift : 학습 데이터 셋과 테스트 데이터셋의 특성이 다른 경우

[1.2 How to Trusting model..?]

위의 2가지 "신뢰" 문제를 해결하기 위해서 논문은 2가지 방법론을 제안한다.
그것은 바로 Lime 과 SP lime 이다.

1.2.1 LIME (locally interpretable model-agnostic explanations)

개요 : 모델의 개별 예측값을 설명하기 위한 알고리즘
: 국지적(local)인 지역에 대해 모델을 설명하는 기법
접근 : 복잡한 모형을 해석이 가능한 심플한 모형으로 locally approximation을 수행하여 설명을 시도.
특징 : 전체 모델이 아닌 개별 prediction의 근방에서만 해석해준다.
: 어떠한 모델이든 적용이 가능하다는 특징이 있다.
: 일반적으로 해석가능한 모델을 해석할 때에는 모델의 구조 및 가중치들을 살펴보면서 신중하게 모델을 해석하지만, LIME 의 경우는 Black box model 에 input 데이터를 넣고 그 결과들을 가지고 해석
: 좀더 자세히 말하자면 input 데이터를 lime 알고리즘에 맞게 변형한 데이터를 black box model 에 여러차례 넣어봄으로서 리턴되는 결과를 가지고 해석하는 방법

1.2.2 SP-LIME [SUBMODULAR Pick]

개요 : 모델 자체의 신뢰 문제를 풀기위해 대표적인 인스턴스를 선택하는 알고리즘

접근 : 모델의 신뢰 문제를 풀기 위해서 논문은 개별 Prediction에 대한 해석을 살펴보는 접근법을 취한다.

: 그런데, 전체 Prediction을 모두 살펴보는 것은 대규모 데이터셋에서 어려운 일이다.

: 따라서 중복되는 정보들을 담고있는 인스턴스들은 제외하고 중요 정보를 담고 있는 소수의 인스턴스를 추려내는 과정이 필요한데, 이를 수행하는 알고리즘이 SP-LIME이다.

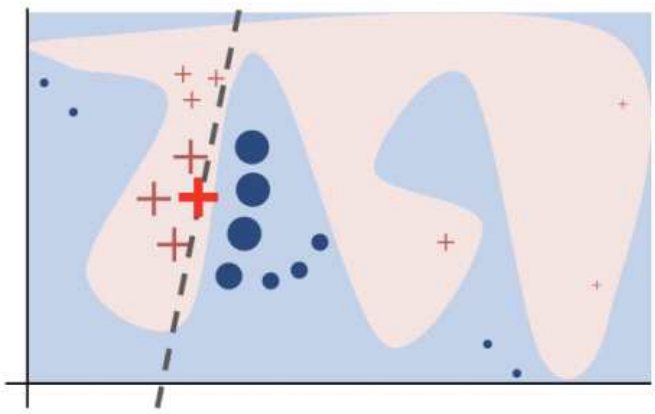
[2. local Fidelity , interpretability]

2.1 Local Fidelity

※ 모델의 충실도(Fidelity) : **모델이 실재와 얼마나 닮았는가**

아래의 전제는 논문의 핵심 전제이므로, 이는 꼭 알아두도록 하자!

『데이터 전체공간에서 모형을 설명하는 것은 어렵지만, 설명하고자 하는 데이터와 가까이 있는 국소적인 공간에서는 의미있는 모형으로 설명할 수 있다.』



EX) 위 예제에서 보면 굵은 빨간 십자가가 설명하고자 하는 데이터라고 하자.
검은 선은 그에 따른 local explanation이다(여기서는 선형회귀로 설명하고 있다.).
전체적으로 볼 때에는 빨간 영역과 파란 영역 구분은 잘 못하지만,
굵은 십자가 근방에서는 선형모델이 데이터의 추이를 매우 잘 설명하고있다는 것을 볼 수 있다.

2.2 Interpretability

※ 모델의 해석력(Interpretability) : 모델이 결과를 예측하는 것을 사람이 얼마나 이해 가능한지

- 국소공간의 설명만은 우리가 원하는 전체공간의 해석을 할 수 없다.
- local 적으로 중요한 변수도 다른공간에서는 중요하지 않을 수 있기 때문이다.
- 전체 공간의 explanation 은 local 적인 공간을 어느정도 설명할 수 있겠지만, 모델이 복잡해지면 이를 해석 (interpretability) 하기가 어렵다.
- 모델이 복잡해질수록 해석력은 낮아진다.

이 둘을 모두 반영할 수 있는 explanation g를 어떻게 찾을 수 있을까?

[3.Definition]

본격적으로 시작하기 전에, 이 논문에서 사용하게 되는 notation 들을 정의하자.

3.1 interpretable explanation

interpretable explanation 은 사람이 이해하는 표현법을 말한다.
데이터의 원 형태를 고려하지 않고, 사람이 해석할 수 있도록 데이터의 모양을 바꾼다.

- ex) text classification에서 binary vector(단어가 있는지 없는지를 나타내는 벡터)
- ex) categorical data는 원핫벡터를 생각할 수 있다.
- ex) image 데이터를 3 chennel 로 나타낸 벡터

original representation, 즉 원 데이터를 $X \in R^d$ 라고 하자.
이때 이 데이터를 해석 가능하게 변환한 데이터 $X' \in \{0,1\}^{d'}$는 binary vector for interpretable representation 라고 생각하자. 즉 해석가능한 표현을 {0,1}들의 집합으로 생각하자는 것이다.

3.2 Functions

g : explanation as a model 즉 우리의 local 데이터를 설명할 때에 사용할 모델
  : g는 g acts over absence/presence of the interpretable components.
  : 모든 g 는 해석하기에 알맞게 쉬운 모델이여야 한다.
G : 해석가능한 모델들의 모음 (linear model , decision tree ....)
domain(g) : $\{0,1\}^{d'}$ (즉 원핫벡터들의 모음)
$\Omega(g)$: measure of complexity (interpretability 와는 반대된다.)
  : 모델이 복잡해질수록 해석력이 낮아지므로 이 값을 낮추고싶다.
  : tree model 에서는 트리의 깊이, linear model 에서는 nonzero weight 계수의 수 등이 될 것
f : explained 될 모델. 이때 $f : R^d \rightarrow R$ 이라 하자.
  : 일반적으로 매우 복잡한 xgboost , deeplearning 모델등이 가능할 것이다.
  : classification에서 f(x)는 probability (또는 binary indicator) 가 될 것이다.
$\pi_x(z)$: proximity measure between instance z to x
  : x,z 의 거리를 재는 측도이다. 어떻게 정할지는 나중에 더 알아보자.
$\mathcal{L}(f,g,\pi_x)$: measure of how unfaithful g is in approximating f in the locality defined by π_x
  : local fidelity를 measure 한다. 이값이 작을수록 local 해석이 좋은 것.
  : 해석가능한 모델 g 가 얼마나 원래 모델 f 와 가깝게 되는지에 대한 측도. 이를 줄이고싶다.

즉 we must minimize $\mathcal{L}(f,g,\pi_x)$ while having Ω(g) be low enough to be interpretable by humans. 이를 수식으로 쓰게 되면 $\xi(x) = \underset{g \in G}{\arg\min}\ \mathcal{L}(f,g,\pi_x) + \Omega(g)$
우리는 interpretablility 와 local fidelity 둘의 균형을 잘 조절하면서 최적화 해야한다. 그 의미가 위의 식에 담겨있다.

[4. SAMPLING for local exploration]

[4.1 how to sampling?]

• 우리는 minimize $\mathcal{L}(f,g,\pi_x)$ 를 할 때에 f에 대한 어떠한 가정 없이 할 것이다. 왜냐하면 모델에 대해 자유로운 해석을 원하기 때문이다.(model - agnostic)

• f에 대한 가정없이 local 에 대한 f의 행동을 관찰하기 위해 우리는 $\mathcal{L}(f,g,\pi_x)$의 근사값을 by drawing sample weighted by π_x를 통해(즉 가중치가 곱해진 샘플들을 통해) f의 local 적인 행동을 분석해보자.

• 우선 x‘ $\in \{0,1\}^{d'}$의 주변에서 sample을 뽑을 뽑는게, local 해석의 첫 단계라고 할 수 있다.

• 여기에서 주변에서 뽑은 sample을 z 라고 하자. 이 때에 sample을 뽑는 경우는 매우 많은데,

1. categorical data set 인 경우

『The R LIME package (Pedersen (2019)) sample with probabilities of the frequency of each category appearing in the original dataset.』

[Limitations of Interpretable Machine Learning Methods 14.1.2.1]

• 즉 우리가 가지고 있는 데이터 셋의 분포대로 sampling을 한다는 의미이다.

• 원래 데이터셋에 korean 의 속성이 있는 데이터가 많았다면 , sampling 할 때에도 koearn 속성의 데이터가 많이 나온다.

2. Numerical features 인 경우

『R LIME package (Pedersen (2019)) for sampling numerical features. The first - and default - one uses a fixed amount of bins. The limits of these bins are picked by the quantiles of the original dataset. In the sampling step, one of these bins will be randomly picked and after that, a value is uniformly sampled between the lower and upper limit of that bin』

[Limitations of Interpretable Machine Learning Methods 14.1.2.1]

• 이 경우는 원래 데이터의 분포를 보고, quantile 에 맞게 bin을 나눈뒤, bin을 random 하게 고른다.

• 그 이후 그 bin 의 upper, lower 값 중에서 uniformly 하게 하나를 고른다.

『Another option would be to approximate the original feature through a normal distribution and then sample out of that one.』

[Limitations of Interpretable Machine Learning Methods 14.1.2.1]

• 이 경우는 N(0,1)에서 샘플을 뽑은뒤 원래 데이터의 평균, 표준편차를 이용해 역변환 한다.

• 즉 원래 데이터가 normal 과 비슷할거라는 약간의 가정이 들어간것

[4.2 Perturbed samples]

• 위에서 만들어진 sample을 perturbed sample $z' \in \{0,1\}^{d'}$ 라고 한다.

• 우리는 이 z'를 모형의 input 형태 $z \in R^d$ 로 바꾸어준다.

• 그리고 모형 f에 z를 input 으로 넣어서 f(z)를 얻는다.

• 이때 f(z) 는 label for the explanation model 로 이용된다.

• perturbed samples 의 집합인 Z 를 가지고 우리는 $\xi(x) = \underset{g \in G}{\arg\min}\ \mathcal{L}(f,g,\pi_x) + \Omega(g)$ 를 최적화 하기 위한 첫 발걸음을 뗀 것이다.

[5. Sparse linear explanation]

[5.1 Find optimization equation]

- 여기서부터 G를 class of linear model 이라고 하고, 식을 전개해보자. 즉 g(z') = $w_g \cdot z'$
- locally aware loss $\mathcal{L}(f,g,\pi_x) = \sum_{z,z' \in Z} \pi_x(z)(f(z)-g(z'))^2$ 로 정의하자.
- g(해석가능)모델이 f(복잡하지만 실제와 비슷함)와 얼마나 가까운지 잰다는 의미에서 $(f(z)-g(z'))^2$ 의 수식은 알맞은 정의이다.
- $\pi_x(z) = \exp(-D(x,z)^2/\sigma^2)$ 로 정의하자.
- D는 distance function 으로서 Text data 일때에는 cosine distance , image 일때에는 L2 Distance 등이 될 수 있을 것이다.
- $\pi_x(z)$를 위 같이 정의하면 가깝다면 높은 값을 가지게 되어, 가까운 데이터를 더 크게 고려하게 된다. 이는 우리가 'local' 적인 해석을 원한다는 점에서 매우 알맞은 정의다.
- 즉 $\mathcal{L}(f,g,\pi_x) = \sum_{z,z' \in Z} \exp(-D(x,z)^2/\sigma^2)(f(z)-g(z'))^2$ 가 된다.
- 이 논문에서 최적화 문제의 모형 복잡도는 $\Omega(g) = \infty 1_{\|\beta\|_0 > k} = \infty I(\|\beta\|_0 > K)$ 로 정의하였다.
- 즉 이는 G를 linear model로 가정했기 때문에 0이 아닌 계수들의 수로 모델의 복잡도를 정의
- 0이 아닌 회귀계수가 K 개보다 많아지면 이 때에는 LOSS 함수가 무한대가 되어버리므로 우리는 K 개 이하의 복잡도를 가진 모델만 고려하게 된다.
- **총 정리하면 최적화문제는** $\xi(g) = \arg\min_{\beta}[\sum_{i=1}^{N} \exp(-D(x,z_i)^2/\sigma^2)(f(z_i)-g(z'_i))^2 + \infty 1_{\|\beta\|_0 > k}]$ **가 된다.**

[5.2 Prove optimization equation]

- 논문에서는 interpretable representation K 개를 선택하는 것은 사용자가 정하도록 하였다.
- 최적의 K 개를 찾는 방법으로 저자는 모델로서 LASSO를 이용하였다.

Algorithm 1 Sparse Linear Explanations using LIME

Require: Classifier f, Number of samples N
Require: Instance x, and its interpretable version x'
Require: Similarity kernel π_x, Length of explanation K

```
Z ← {}
for i ∈ {1, 2, 3, ..., N} do
    z'_i ← sample_around(x')
    Z ← Z ∪ ⟨z'_i, f(z_i), π_x(z_i)⟩
end for
w ← K-Lasso(Z, K)  ▷ with z'_i as features, f(z) as target
return w
```

- 계속 Z를 업데이트하는데, 이때에 (z_i'(x데이터) , $f(z_i)$(y값), $\pi_x(z_i)$가중치) 세 개가 Z 에 추가되게 된다.
- 여기서 선택한 g 모델(LASSO)는 z_i'데이터들을 x변수로, $f(z_i)$ 들을 각각의 y 변수로 고려한 상태에서 K개의 설명변수를 가지기 위해 λ를 조절한다. 이러면 return 으로 K개의 변수를 가지고 있는 LASSO 모델을 내보내게 된다.
- 선택된 K 개의 변수 : 다른 변수보다 중요하다고 판단된 K 개의 변수
- 각 변수들의 계수 : lasso 모델의 계수이므로 이는 Importance를 의미한다.

(+0.3 이면 그 변수가 있을 때(1일 때) 결과값에 0.3 이 더해진다. 의 의미)

[6. SUBMODULAR Pick for explaining models]

SP-LIME : 모델 자체의 신뢰 문제를 풀기위해 대표적인 인스턴스를 선택하는 알고리즘이다.

[6.1 Pick Problem]

- 데이터 X $\in R^{n*d}$ 를 고려하자.
- 모든 데이터를 살펴볼 수 없기 때문에, 특징이 다르고 중요한 데이터만을 뽑아서 생각한다.
- 살펴볼 데이터 budget B를 정한다. 이는 데이터를 총 몇 개 살펴볼지의 값이다.
- original representataion 으로부터, 해석가능한 표현 $X' \in R^{n*d'}$ 를 만든다.
- 데이터의 local 한 importance를 나타낼 matrix W를 정의한다.
- 이때 linear model을 explanation 으로 사용했다면 데이터 x_i에 대하여 explanation $g_i = \xi(x_i)$ 를 얻을 수 있고, $W_{ij} = |w_{g_{ig}}|$ 로 정의할 수 있다.
- 추가적으로 I_j (j column 의 global importance를 정의) $I_j = \sqrt{\Sigma_i W_{ij}}$ 로 정의한다.
- 최대한 특징이 다르며 많은 정보를 포함하고 있는 데이터를 추출하기 위해서

$c(V,W,I) = \sum_{j=1}^{d'} 1_{\exists\, i \in V:\, W_{ig} > 0} I_j$ 라고 coverage function을 정의한다.

ex)

위 matrix 는 W를 나타낸다. 그림을 보면 row 는 데이터, f 는 feature 이다.
맨 위 데이터부터 x1,x2,x3,x4,x5 라고 하자.
각 검은 사각형 안에 1 값이 들어있다면 V = {1,4} 인 경우에는 c(V,W,I) 는 (1+4) + (4+2) 가 된다.

- 위의 방법처럼 일부 데이터를 선택하는 방법을 Pick problem 이라고 하고 , 이 방법은

$PICK(W,I) = \underset{V, |V| \le B}{\arg max}(c(V,W,I))$ 가 된다.

[6.2 SP - Lime]

- 하지만 이 경우 수많은 V를 고려해야하기 때문에 매우 어렵다. 그러므로 이를 greedy 하게 풀어내는 방법이 차선책이 될 수 있을 것이다.
- 그 방법이 SP - Lime 이다.

Algorithm 2 Submodular pick (SP) algorithm

Require: Instances X, Budget B
  for all $x_i \in X$ **do**
    $\mathcal{W}_i \leftarrow \textbf{explain}(x_i, x_i')$ ▷ Using Algorithm 1
  end for
  for $j \in \{1 \dots d'\}$ **do**
    $I_j \leftarrow \sqrt{\sum_{i=1}^{n} |\mathcal{W}_{ij}|}$ ▷ Compute feature importances
  end for
  $V \leftarrow \{\}$
  while $|V| < B$ **do** ▷ Greedy optimization of Eq (4)
    $V \leftarrow V \cup \text{argmax}_i \, c(V \cup \{i\}, \mathcal{W}, I)$
  end while
  return V

• 위 식을 보면 각각의 setep 마다 $c(V, W, I) = \sum_{j=1}^{d'} 1_{\exists\, i \in V:\, W_{ig} > 0} I_j$ 를 최대화 하기 위하여, sample을 선택하고 있다.
• 총 B 개 고르고 나면 알고리즘이 종료되고, 그에따라 나온 V 가, 우리가 모델을 평가할 때에 이용할 대표적인 인스턴스 이다.

[7. Pros and Cons]

[장점]
- LIME은 Tabular, Text, Images 에서 모두 사용할 수 있는 거의 없는 방법 중 하나이다.
- 특히, embedding한 data와 같이 추상화된 data에도 적용할 수 있다. (그 '결과' 만 이용하니까)

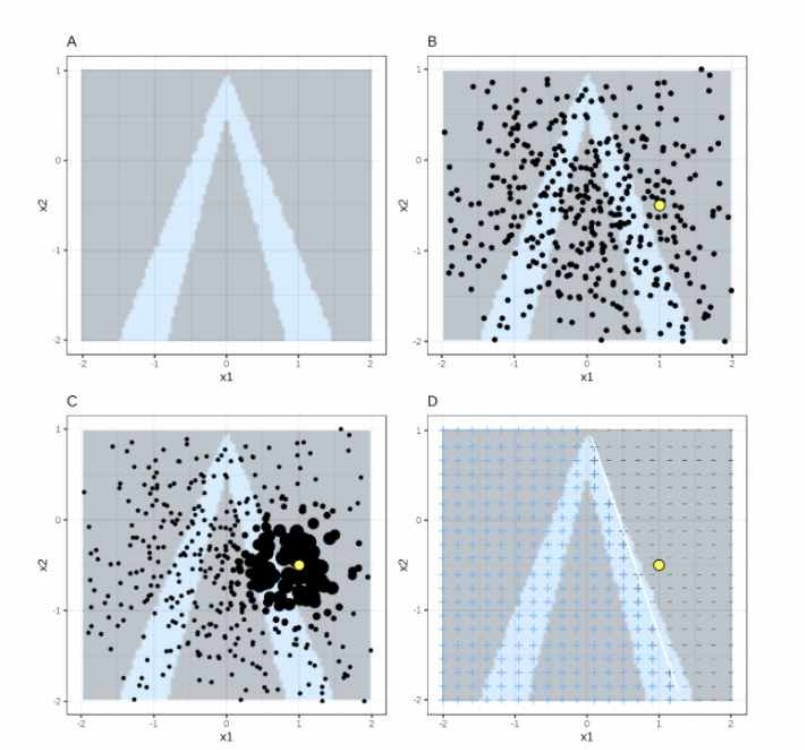
[단점]
- Lime 의 샘플링 data는 Gaussian distribution에 의해 sampling 되는경우 이는 feature간의 correlation을 무시하게 됨
- 설명 모델의 복잡성을 사용자가 미리 정의해야한다. (어느정도가 적당한지도 모른 채..)
- 설명력의 불안정성이 큰 문제 → sampling을 계속해서 돌리다보면 돌릴때마다 모델의 설명이 계속 바뀐다.

[Reference]

https://rstudio-pubs-static.s3.amazonaws.com/442427_4d8243a35bcd4df3982b992d851a0a6b.html
https://compstat-lmu.github.io/iml_methods_limitations/lime.html
https://arxiv.org/pdf/1602.04938.pdf
https://towardsdatascience.com/understanding-how-lime-explains-predictions-d404e5d1829c
https://dodonam.tistory.com/202

[※ Examples]

[Example 1]



- 위 값은 lime 의과정이다
- 파란색은 0, 검정색은 1 의 class를 가지는 경우
- 설명할 예측 데이터셋은 노란색이다. 그 주변으로 생성된 데이터가 검은색이다.
- 설명을 위해 예측할 가까운 sample 에 높은 weight를 매긴다. 그에따라 sample 주위의 데이터가 큰 점이 된 것을 볼 수 있다.
- 그 local에서 해석된 경우가 마지막 경우이다. + 는 파란색(0) , - 는 검은색(1)을 예측한 것
- local 에서는 잘 맞아보이는듯하다.

[Example 2]

LIME for Text

Text data를 분석할때는 모델 해석을 위한 dataset을 구성하는 방법이 tabular data와는 다름
Text LIME은 dataset을 생성할때 Original TEXT에서 단어(word)를 제거해가며 분석하는 방식으로 dataset을 만듦
예를 들면, 아래 표와 같이 각 단어 유무에 따라 1, 0 으로 표시한 data를 만듦

	CONTENT	CLASS
267	PSY is a good guy	0
173	For Christmas Song visit my channel! ;)	1

The next step is to create some variations of the datasets used in a local model. For example, some variations of one of the comments:

	For	Christmas	Song	visit	my	channel!	;)	prob	weight
2	1	0	1	1	0	0	1	0.17	0.57
3	0	1	1	1	1	0	1	0.17	0.71
4	1	0	0	1	1	1	1	0.99	0.71
5	1	0	1	1	1	1	1	0.99	0.86
6	0	1	1	1	0	0	1	0.17	0.57

여기서, prob은 해당 데이터를 black box model에 넣었을 때의 return된 확률값을 의미함
weight의 경우, original data에 비해 소실된 데이터의 비율로 weight를 나타냄
예를 들어 전체 7단어에서 한단어가 빠진 6단어로 표현했을 때,
weight = 1 - 1/7 로 나타낼 수 있음
그리고 이렇게 만들어진 data를 LIME Algoritm에 집어 넣어 아래 표처럼 각 feature weight를 계산하게 됨

case	label_prob	feature	feature_weight
1	0.1701170	good	0.000000
1	0.1701170	a	0.000000
1	0.1701170	is	0.000000
2	0.9939024	channel!	6.180747
2	0.9939024	For	0.000000
2	0.9939024	;)	0.000000

The word "channel" indicates a high probability of spam. For the non-spam comment no non-zero weight was estimated, because no matter which word is removed, the predicted class remains the same.

이런 식으로 나오면 channel이라는 단어가 spam을 구분하는데 큰 weight가 매겨져 있음을 확인할 수 있음

[Example 3]
LIME for Images

Image를 위한 LIME은 tabular, text 데이터와는 또 다르게 데이터셋을 생성시킴
일반적으로 각 pixel을 random하게 변형시킬 것이라고 생각하지만 그렇게 하지 않음
superpixels을 껐다 켰다하는 식으로 데이터를 생성 시킴
superpixels : pixels 끼리 서로 비슷한 색으로 연결되어 있는 것을 지칭
superpixels을 끈다는 의미는 superpixel의 값을 사용자가 정의한 값으로 변경 시킴을 의미 (gray)
Inception v3 model을 이용해 bread를 구분하는 문제가 있다고 하자

FIGURE 5.36: Left: Image of a bowl of bread. Middle and right: LIME explanations for the top 2 classes (bagel, strawberry) for image classification made by Google's Inception V3 neural network.

LIME은 위의 그림처럼 label을 판단하는데 영향을 주는 부분을 표시할 수 있음
여기서 green은 해당 pixels로 인해 해당 label을 판단하는데 양의 가중을 주는 것을 의미하고, red는 음의 가중을 주는 것을 의미함 (해당 label 이 아님을 증거)
위 그림에서 베이글을 예측하는데 그림의 녹색부분이 큰 역할을 했다는 것을 바로 확인할 수 있음