# Contents

# 맥주 추천 NLP모델

- 사용 언어


(Keras, Numpy, Pandas)

- 사용 알고리즘
  - Word2Vec에 기반한 워드 임베딩
  - LSTM with attention mechanism

- Work Flow(담당 역할은 강조 표시)

- 데이터 수집(크롤링)

①**인스타그램**

②다음 까페

③ratebeer

**- 데이터 양식 통합 및 데이터 전처리**

① Word2Vec 사전 훈련

② 이모티콘 제거

**- 모델 작성 및 적합**

**① 1단계 모형 : 단순 LSTM**

**② 2단계 모형 : LSTM with attention Mechanism**

**③ 워드 임베딩에 기반한 이상형 월드컵**

- 서버 구축 및 서비스화

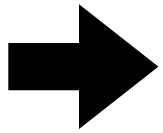
① start-polling 방식 서버 구축

② 텔레그램 챗봇 서비스화

# Summary



| Word Embedding | CNN | LSTM |
| --- | --- | --- |
| 각 단어를 비슷한 의미에 따라 벡터공간에 비슷한 위치에 사영시킨다. | 문장에서 각 단위별로 뚜렷한 특성만을 추출해 압축한다. | 입력 받은 특성값을 앞 뒤, 시계열적인 문맥을 고려하여 의도를 추측하고 학습하는 시계열 학습 알고리즘이다. |

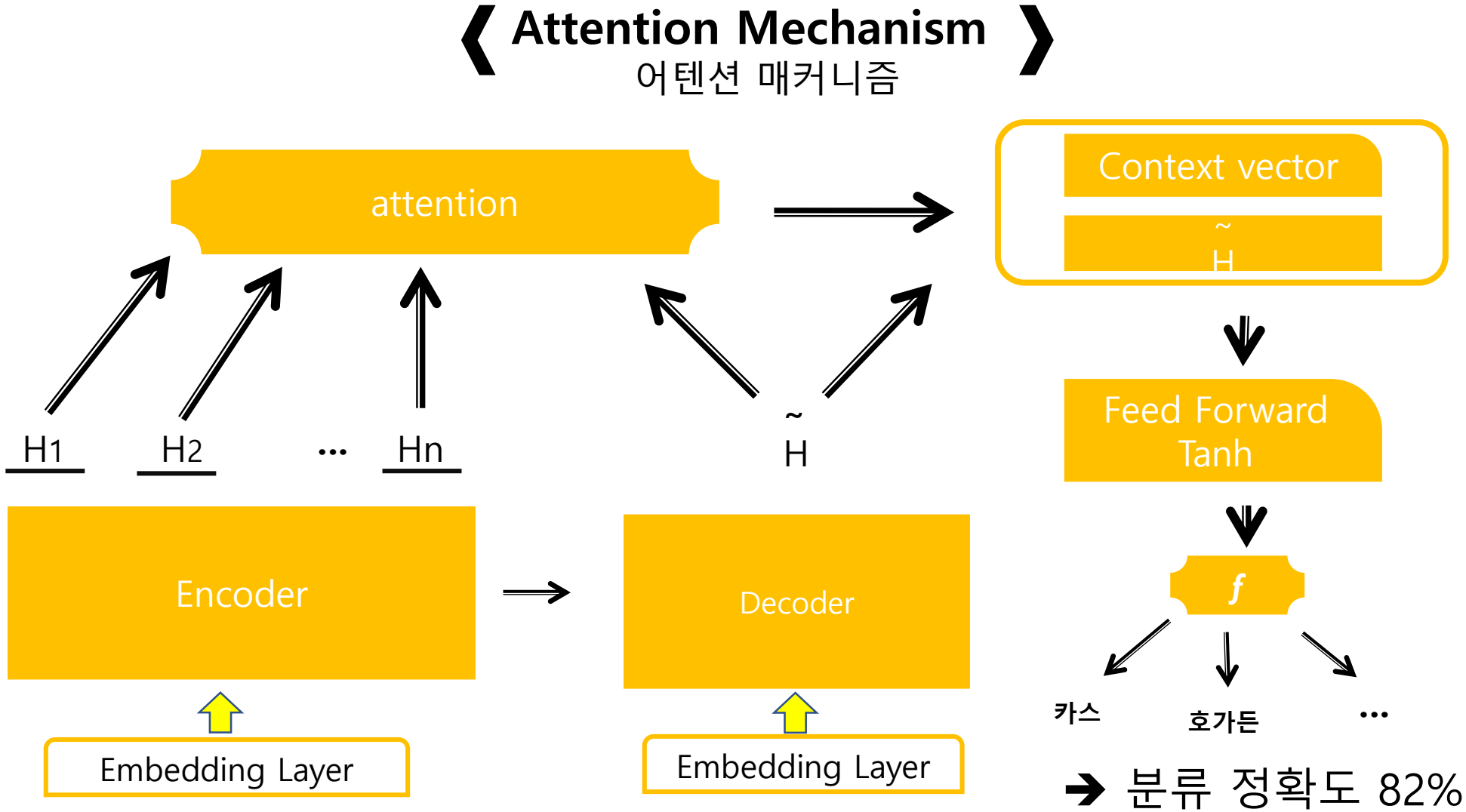


**Chatbot Model 1**

분류 정확도 48~52%

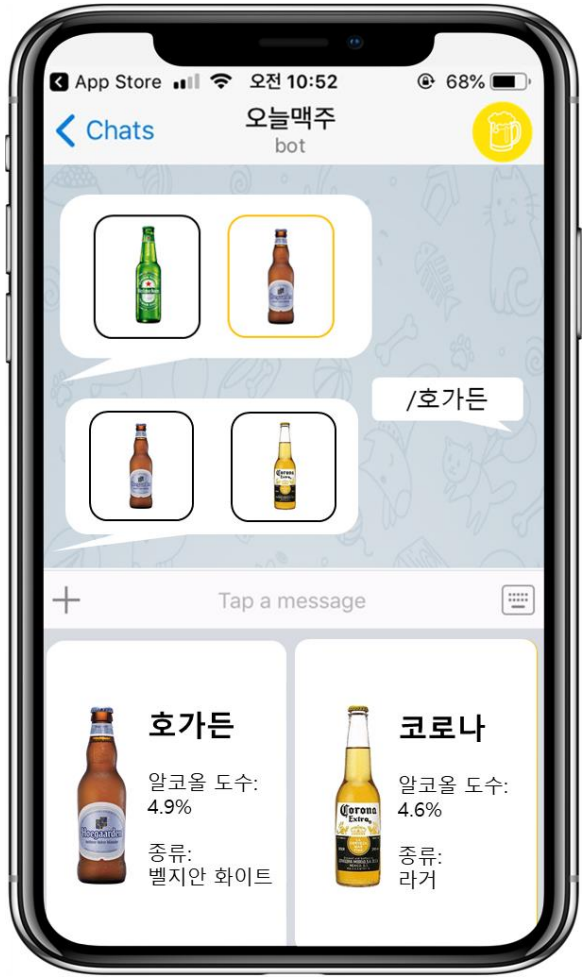
각 맥주 사이의 단어 분포 차이가 뚜렷하지 않으면
**분류 정확도**가 떨어지는 문제 발생

# Summary

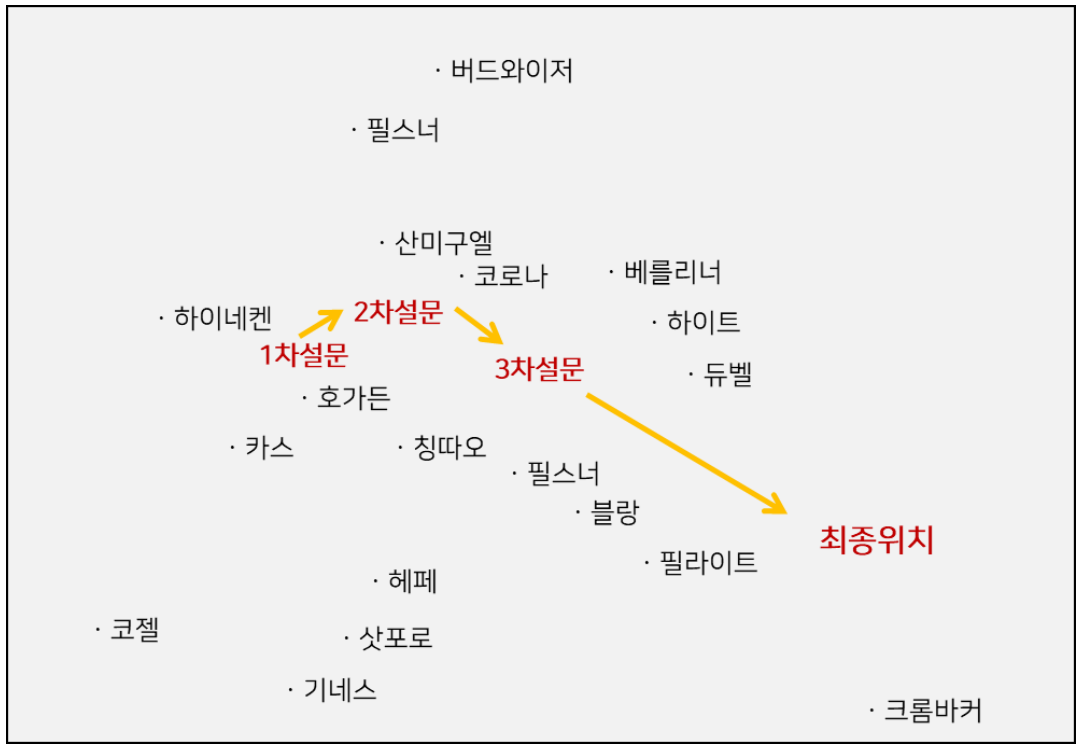



➔ 분류 정확도 82%까지 상승

# Summary




Word Embedding




유저의 단계별 맥주 선택으로
최종 선호도에 대한 정보를 데이터 베이스에 저장한다.

➔ 맥주 추천 시 유저의 선호도를 반영하는 데에 사용.

# 구현상 어려운점과 극복과정

- **맥주라는 상품 특성상 차이가 뚜렷하지 않음**

- 인스타그램, 맥주 리뷰 사이트, 다음 까페등에서 자연어 데이터를 **10만건 정도 크롤링**하여 학습 데이터로 활용
- 다만 맥주는 **맛의 특성 차이가 뚜렷하지 않았고**, 모형이 패턴을 제대로 학습하지 못함. 1단계로 적합한 단순 LSTM모형에선 예측 정확도가 50%에 불과한 문제점이 발생

# 구현상 어려운점과 극복과정

- **Attention-mechanism 구현의 어려움**

- 특성 차이가 뚜렷하지 않은 문제를 해결하기 위해 언어번역 분야에서 광범위하게 쓰이는 attention-mechanism을 고려
- 역전파 과정에서 **최대 두번의 학습 기회를 갖지만**, 언어 번역에만 최적화된 코드들만 있어 **챗봇용으로 새로 구축할 필요**가 있었음
- 논문에 의존하여 **Keras의 저수준 API**로 직접 구축. 행렬의 차원이 맞지 않거나 버그로 인하여 알 수 없는 오류가 발생하는 등 **소스코드를 직접 수정해가면서 모델 작성**
- **최종적으로 val-acc 82% 수준의 모델 작성**

# 구현상 어려운점과 극복과정

- **문장 판독과 소비자 취향의 적절한 Trade-off**

- 문장은 소비자의 **현재 상태를 나타내는 데이터**에 불과. '기분 꿀꿀한데 좋은 맥주 추천해줘 ' 란 문장은 소비자의 현재 상태만 나타남
- **소비자의 근원적인 취향을 고려**한 답을 제시할 필요가 존재
- 워드 임베딩된 단어 벡터를 활용, Attention-mechanism의 Decoder에 각 맥주의 단어 벡터를 주입하여 Encoder의 문장 입력과 Decoder의 맥주 단어 벡터간 연관도를 학습
- 소비자가 이상형 월드컵을 실시하면 **해당 유저의 취향이 취향 벡터로 저장**, 예측시에 입력 문장과 함께 이를 함께 고려

# 방한관광객수 예측 모델

- 사용 언어

(arima, ggplot, dplyr)

- 사용 알고리즘
  - ARIMA(파라미터 적합)
  - Box-Ljung(최적 적합 판단)

- Work Flow

- **데이터 수집**

**①관광지식정보시스템(방한관광객수)**
**②일본 경제산업성(일본 산업생산지수)**
**③한국은행 환율 통계(원-엔 환율)**

**- 탐색적 데이터 분석**

**- 데이터 전처리**
**① Time-Table로 데이터프레임 변환**
**② 모델 적합용 투입변수 예측값 생성**
**③ 수치 예측용 투입변수 예측값 생성**

**- 모델 작성 및 적합**
**① 파라미터 적합 : ARIMA**
**② 예측값 생성 : 회귀분석(lm)의 predict 함수**

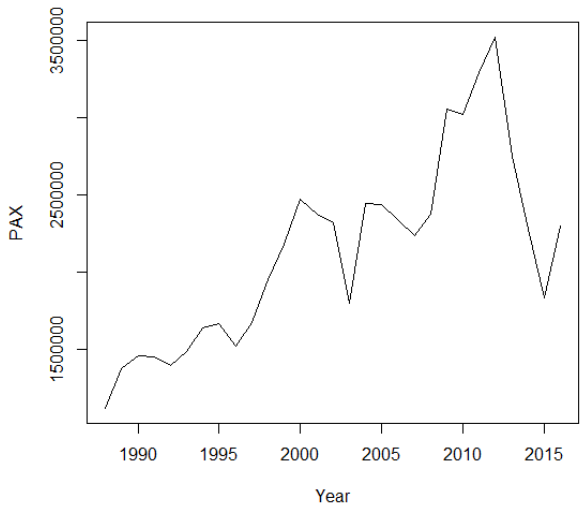
# Summary

방한관광객수 예측 모델

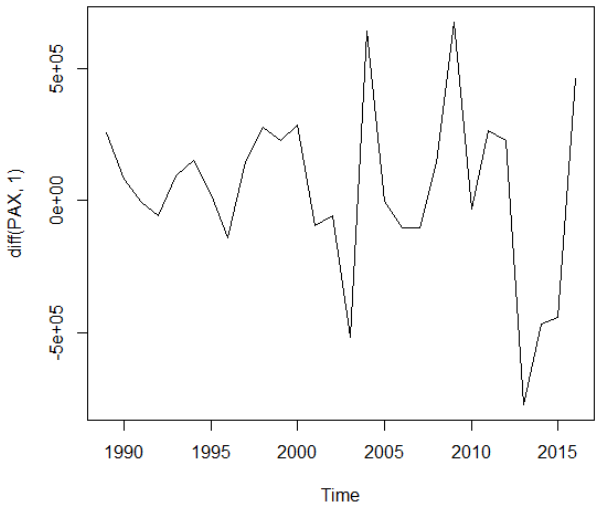
< 데이터셋 >

| | year | PAX | IAA | Fri_ab | Fri_almost | Cur |
|---|---|---|---|---|---|---|
| 1 | 1988 | 1124149 | 85.72 | 12.2 | 38.7 | 570.56 |
| 2 | 1989 | 1379523 | 90.21 | 7.6 | 33.1 | 487.51 |
| 3 | 1990 | 1460291 | 94.39 | 9.6 | 33.1 | 491.58 |
| 4 | 1991 | 1455090 | 96.98 | 10.7 | 32.4 | 545.95 |
| 5 | 1992 | 1398604 | 95.43 | 8.5 | 34.4 | 617.08 |
| 6 | 1993 | 1492069 | 95.59 | 8.9 | 34.5 | 725.67 |
| 7 | 1994 | 1644097 | 96.64 | 7.9 | 34.0 | 787.91 |
| 8 | 1995 | 1667203 | 97.99 | 7.6 | 34.6 | 824.45 |
| 9 | 1996 | 1526559 | 100.19 | 6.6 | 29.2 | 739.59 |
| 10 | 1997 | 1676434 | 101.17 | 6.2 | 32.7 | 784.02 |
| 11 | 1998 | 1954416 | 99.11 | 10.9 | 36.9 | 1074.41 |
| 12 | 1999 | 2184121 | 99.68 | 12.8 | 35.5 | 1048.64 |
| 13 | 2000 | 2472054 | 101.40 | 14.8 | 36.6 | 1048.92 |
| 14 | 2001 | 2377321 | 100.84 | 11.9 | 28.4 | 1062.41 |
| 15 | 2002 | 2320837 | 100.38 | 13.5 | 40.5 | 999.57 |
| 16 | 2003 | 1802542 | 101.27 | 13.7 | 41.3 | 1029.76 |
| 17 | 2004 | 2443070 | 102.92 | 9.0 | 46.5 | 1058.76 |
| 18 | 2005 | 2440139 | 104.62 | 12.5 | 38.7 | 930.66 |
| 19 | 2006 | 2338921 | 106.57 | 12.3 | 35.9 | 821.49 |
| 20 | 2007 | 2235963 | 107.51 | 14.9 | 39.9 | 789.75 |
| 21 | 2008 | 2378102 | 105.55 | 13.3 | 43.8 | 1076.63 |

<추세성의 제거>




차분 전

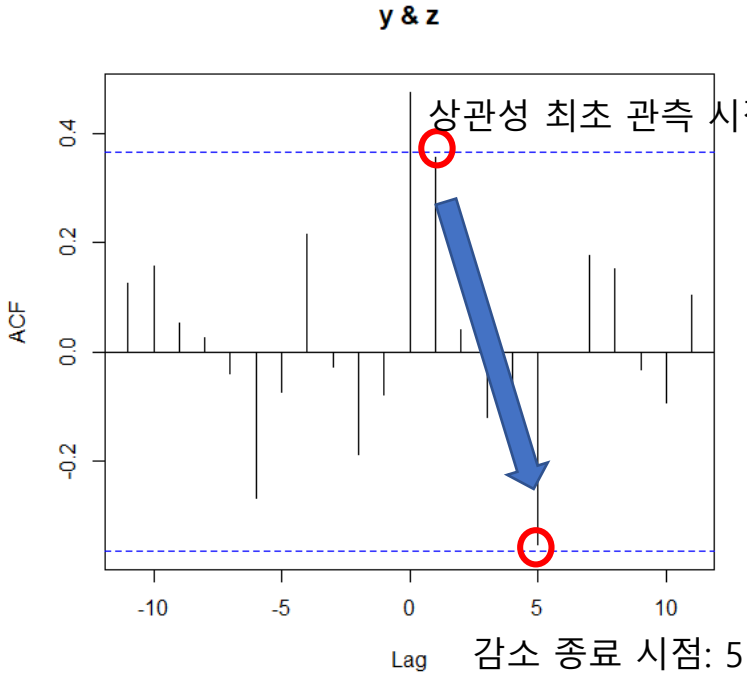



차분 후

# Summary

방한관광객수 예측 모델

< 교차상관함수 그래프 >

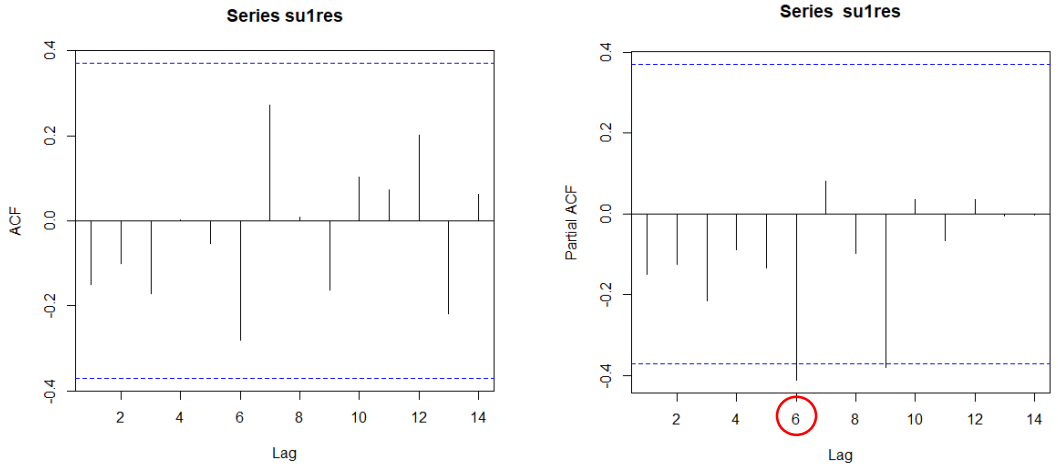



상관성 최초 관측 시점 : 0

감소 종료 시점: 5 or 1

지연모수 : 0, 투입모수 : 1, 산출모수 : 5 or 1

<가적합 후 여과된 잔차의 ACF , PACF>




ARMA(6,0) 모형을 고려

# 구현상 어려운점과 극복과정



- **회귀분석 모형 사용의 제한**

- 프로젝트 초기 단계에선 중회귀분석 방법론을 사용, 적합을 시도
- 파라미터는 유의한 것으로 나왔으나 오차가 심각. 원인 분석 결과 결과변수(방한관광객수)에 **자기상관성이 존재**. 계수에 대한 **t검증을 신뢰할 수 없게됨.**
- 회귀분석에서 시계열분석으로 방법론 전환

- **시계열 분석에 대한 지식**

- 적용보다 시계열 분석 방법론 자체에 대한 지식이 부족
- ARIMA를 적용하기로 하고, 이를 위해 **시계열 분석 전공서적**과 **20여개의 논문**으로 3개월 동안 학습

# 코스타리카 빈곤선 예측 모델

- 사용 언어


(Scikit-learn ,Numpy, Pandas)

- 사용 알고리즘

- SMOTE (타겟 불균형 해소 업샘플링)
- xtreeRegressor (결측값 대체값 생성)
- LightGBM (모델 적합)

- Work Flow(담당 역할은 강조)

- 과업정의(그룹 스터디)
① 기존 분석방법론(Proxy) 탐구
② 코스타리카 현재 상황 탐구
③ 변수의 의미에 대한 분석

**- 탐색적 데이터 분석**

**- 결측 처리 및 데이터 전처리**
**① 대체값 생성(xtreeRegressor)**
**② 업샘플링(SMOTE)**

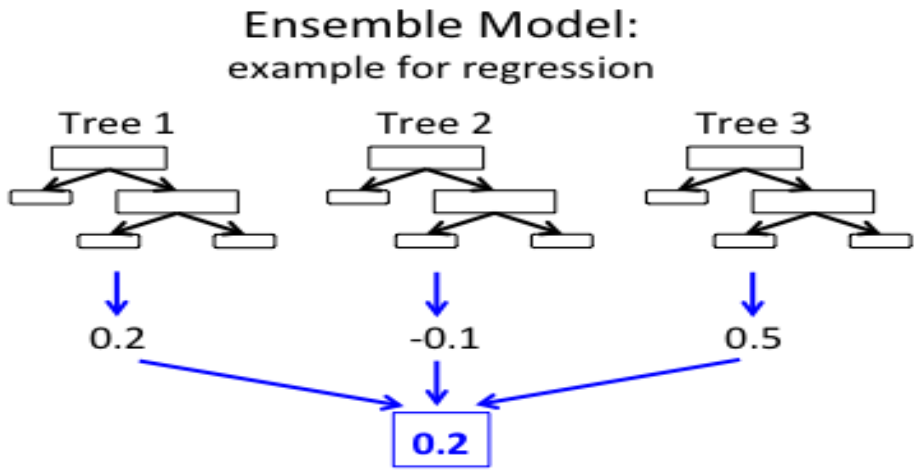
**- 모델 적합**
**① 하이퍼파라미터 최적화(그리드 서치)**
**② LightGBM 적합**

# Summary

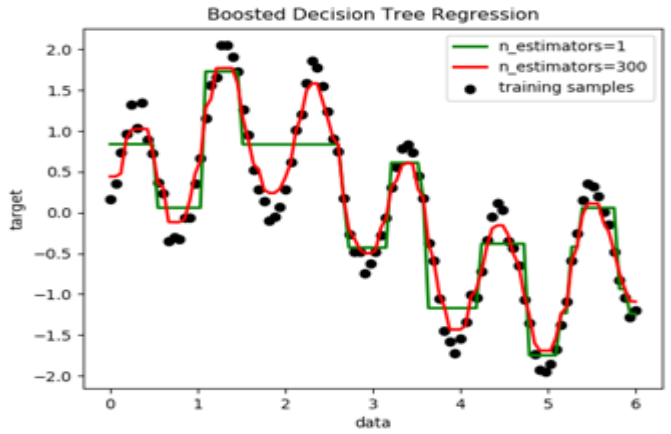
코스타리카 빈곤선 예측모델

- 결측값의 처리

- XtreeRegressor를 이용해 특성들이 비슷한 것으로 분류된 샘플들의 **평균값으로 결측값을 예측**한다.
- 앙상블 기법에 기반한 트리모형의 회귀예측은 **여러 약분류기(트리)들의 동일 그룹 평균값의 평균값**을 토대로 예측치를 생산한다.



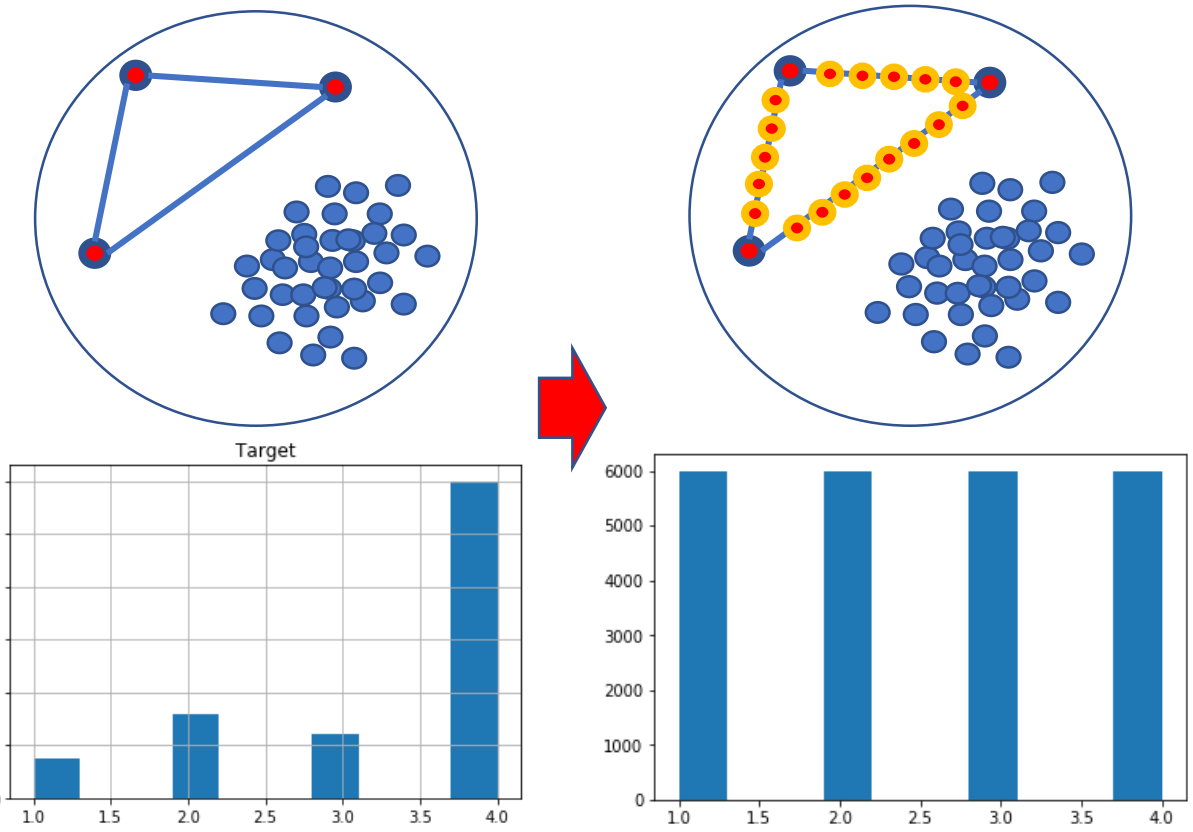

<앙상블 모형의 회귀 예시>




<회귀선>

# Summary

- 타겟 불균형의 처리



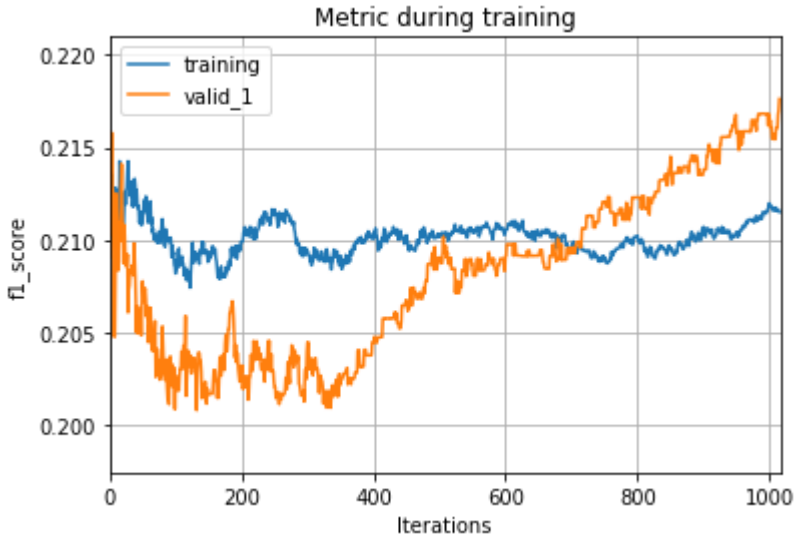

SMOTE 알고리즘을 통해 업샘플링(Up-sampling) 실시

# Summary

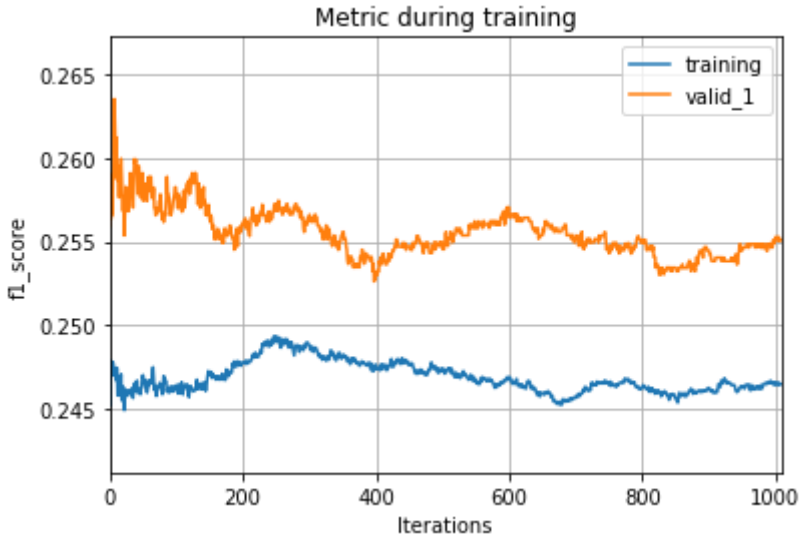
코스타리카 빈곤선 예측모델

- 모델 적합

<원본 데이터셋>




<업샘플링 데이터셋>




f1_score 0.5점 상승 확인, Kaggle 최다점 0.410점에 근접

# 구현상 어려운점과 극복과정



- **결측값의 처리**

- 몇몇 변수들에서 결측값이 발생
- 이 결측값이
 ①**데이터 부족으로 인한 누락 사항**인지,
 ②**어떤 의도를 갖고 기재하지 않은것**인지 불분명

- 데이터셋 구조를 통해 결측값의 의미를 추리
 ① 데이터 부족으로 인한 누락의 경우 **회귀트리를 통해 결측값을 예측**
 ② 0을 결측값으로 처리한 경우엔 **0으로 일괄 처리**

# 구현상 어려운점과 극복과정



- **도메인 지식의 부족**

- 코스타리카 현지의 경제 사정과 빈곤 상황에 대한 **도메인지식이 부족**
- 프로젝트를 함께 진행하는 사람들과 **공동 스터디**를 통해 과업을 이해하고 **데이터 전처리에 적극 활용**함

- **친절하지 않은 변수명**

- 스페인어권인 코스타리카 특성상 변수명도 스페인어로만 간략하게 기재
- 변수에 대한 특별한 설명이 없어서 무엇을 조사한 변수인지, 왜 이런 파생변수를 만들었는지 이해하기 어려움
- 공동 스터디를 통해 변수를 이해하고**, 데이터 전처리에 적극 활용함**

# 예방가능 사망자수 예측 모델

- 사용 언어

(ggplot, dplyr)

(pandas, numpy)

- 사용 알고리즘

- 푸아송 회귀분석

- Work Flow

- 데이터 수집

① 경찰청(연도별 고속도로별 사망자 수)

② 한국도로공사(연도별 고속도로별 평균속도)

③ 통계청(연도별 실업률)

- 탐색적 데이터 분석

① 연도별 고속도로별 평균속도 확인(aggregate)

② 부트스트랩(평균속도의 평균 확인)

- 데이터 전처리

① 평균속도 데이터셋 통합(10만 point)

② 월별 평균속도를 연도별 평균속도로 변환

-모형 적합

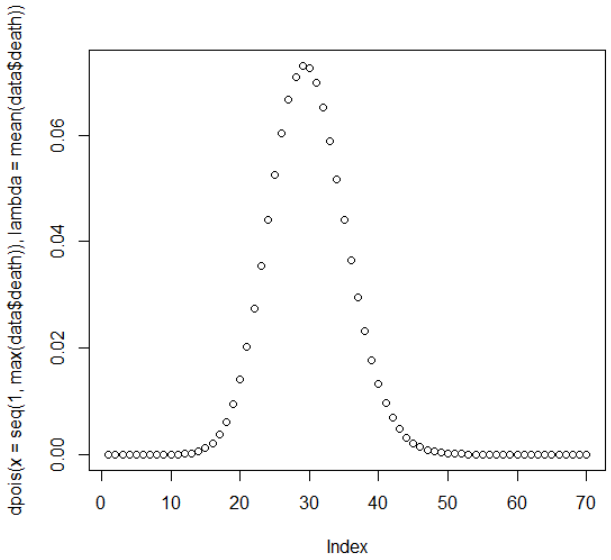
① 푸아송 회귀분석

② 과산포 여부 확인

# Summary

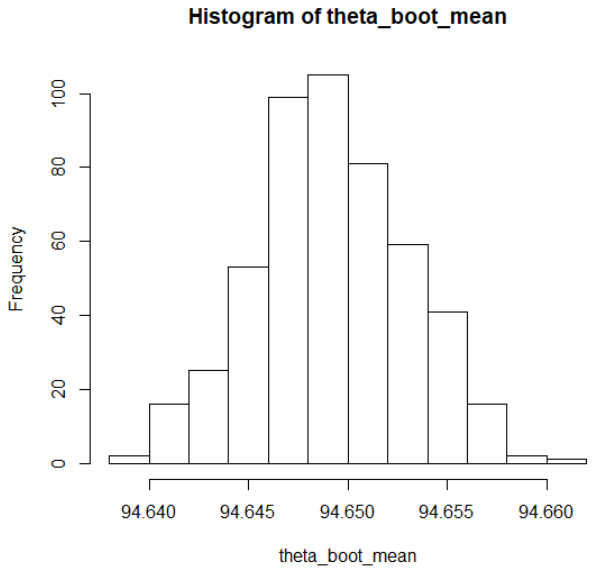
예방가능 사망자수 예측 모델

<연도별 사망자수 확률분포(예상)>




Lambda = 29
X = seq(1,70)

<평균속도의 부트스트랩>




Mean = 94.65

# Summary

예방가능 사망자수 예측 모델

<모델 적합 결과(과산포 조정 이전)>

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -36.27809    1.96445 -18.467  < 2e-16 ***
speed         0.11138    0.02092   5.325 1.01e-07 ***
year         -0.06577    0.02282  -2.883  0.00395 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 73.996  on 23  degrees of freedom
Residual deviance: 40.582  on 21  degrees of freedom
AIC: 169.45
```

<모델 적합 결과(과산포 조정 이후)>

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -36.27809    2.79774 -12.967  < 2e-16 ***
speed         0.11138    0.02979   3.739 0.000185 ***
year         -0.06577    0.03249  -2.024 0.042972 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 2.02829)

    Null deviance: 73.996  on 23  degrees of freedom
Residual deviance: 40.582  on 21  degrees of freedom
AIC: 169.45
```

# 구현상 어려운점과 극복과정



- 데이터셋 통합

- 한국도로공사에서 제공하는 평균속도 데이터셋은 연도별이 아니라 월별로, 또 모든 고속도로를 하나의 데이터셋에 통합하여 제공
- 월별로 분할 압축되어 있는 데이터들을 하나의 데이터 프레임으로 통합하여 분석 대상 고속도로의 연도별 데이터로 재가공할 필요성
- 파이썬의 os모듈을 사용해 하나의 폴더로 통합시킨 후, 폴더의 목록을 불러와 이 목록을 토대로 csv파일을 하나씩 임포트, 데이터프레임에 통합하는 방식으로 코드를 작성