딥러닝 기반 핵심 산업별 빅데이터 분석

<머신러닝&딥러닝 파일럿 프로젝트>

주 제	Costa Rican Household Poverty Level Prediction	링 크	https://www.kaggle.com/c/costa-rican-household-poverty-prediction
팀 명	원픽	일 자	2018년 11월 29일
팀 장	문정연<jymoon1115@gmail.com>	팀 원	공정배, 이진수

1. 과제 개요

1) 목적 : 사회 복지 지원이 가장 필요한 세대 확인

- 사회적 필요성 존재

적절한 복지 예산 집행을 위해서는 소득 자격 확인 알고리즘이 매우 중요하다.

- 국가적 특성을 고려한 배경

라틴 아메리카에서 주로 사용되는 PMT(Proxy Means Testing) 방법은 주택의 벽의 재질, 소유하고 있는 가전제품 종류 등의 가시적인 평가지표를 활용해 모델을 만들어 적용한다. 그러나 여전히 모델의 정확성을 높일 필요가 있다. 이에 따라 기존 평가 지표 외의 다양한 코스타리카의 가계 특성에 대한 데이터 세트를 기반으로 보다 적합한 모델을 찾아내고자 한다.

2) 분석방법 : 텐서플로우를 활용

- TRAINING 데이터를 학습
- TEST 데이터의 경제 수준을 1-4의 4가지 단계로 구분

1	2	3	4
극빈빈곤	중등빈곤	취약 가구	비취약 가구

3) 의의

- 학습적 의의 : 텐서플로우의 다중분류기법을 적절히 활용
- 실용적 의의 : 사회적으로 꾸준히 논의가 이루어지고 있는 데이터를 분석해 올바른 예산 집행에 활용 가능

2. 데이터 설명

1) train data & test data

한 행은 데이터 샘플에서 한 사람을 나타냄, 여러 사람이 한 가구에 속할 수 있으며 가구원 수에 대한 예측만 기록됨.

2) 주요 필드 소개

열 이름	열 설명
Id	각 행의 고유 식별자.
Target	수입 수준의 그룹을 나타내는 서수 변수 (궁극적으로 알고자 하는 값)

idhogar	각 세대의 고유 식별자, 세대 전체의 특징을 생성하는 데 사용됨
parentesco1	이 사람이 세대주인지 여부를 나타냄
총 열 개수	142개

3. 과제 수행 내역

1) 피쳐값 분석 : 시각화를 통한 피쳐 단순화

- 타겟 데이터 분석

우리가 맞추고자 하는 궁극적인 결과값 y의 분포를 분석해 차후 결과값과 비교에 활용, 4(비취약 가구)가 가장 많음을 알 수 있음

- 세대주만 추출해 train데이터 가공

Idhogar은 세대 식별 고유 값으로 같은 행끼리는 같은 가구원임을 파악할 수 있다. 같은 세대임에도 불구하고 TEST 최종 값이 다르게 나타나는 경우가 있으며 이러한 오류를 줄이기 위해서 세대주만을 뽑아 분석을 진행

- corr 분석을 통한 피쳐 단순화

크게 유의미할 것으로 보이지 않는 데이터 열은 피쳐로 선택하지 않으며, 동일한 내용을 담고 있는 열은 각각의 피쳐값을 척도로 파악해 통일하여 하나의 피쳐로 만들어냄

- TSNE 기법을 활용해 피쳐 단순화 성공여부 파악

초기 모든 변수 추출하였을 때와 피쳐를 단순화 했을때를 시각적으로 비교하기 위해 2D, 3D그래프로 구현

2) 가중치 표 그리기

Sorting을 통해 가중치가 큰 것과 작은 것을 파악해 영향을 많이 미치는 요소를 보고자 함, 각각의 빈곤 유형에 따라 영향을 가장 많이 미치는 요소들이 다르게 존재한다는 사실을 파악 가능

3) 머신러닝 과정

요인	우리가 발견한 최적화 모형 변수
활성화 함수	렐루 함수
Hidden layer 수	9
node	64
Optimizer 방식	아담 경사 하강법
정확도 개선	Xavier initialize
오차함수	softmax_cross_entropy_with_logits
Step	5000

4. 결과 보고

1) Train 데이터에 대한 모형 정확도 파악 : Loss: 0.421 / Acc: 85.60%

loss가 감소하고 accuracy가 증가하는 안정적인 형태를 보임

```
In [55]: # Launch the graph in a session.
sess = tf.Session()

# Initializes global variables in the graph.
sess.run(tf.global_variables_initializer())

for step in range(5001):
    sess.run(optimizer, feed_dict={X: x_data, Y: y_data, keep_prob: 0.7})
    if step % 1000 == 0 or step < 100:
        loss, acc = sess.run([cost, accuracy], feed_dict={
                             X: x_data, Y: y_data, keep_prob: 0.7})
        print("Step: {:5}, Wt Loss: {:.3f}, Wt Acc: {:.2%}".format(
            step, loss, acc))
```

```
Step:    85,    Loss: 1.113,    Acc: 62.70%
Step:    86,    Loss: 1.091,    Acc: 62.97%
Step:    87,    Loss: 1.087,    Acc: 63.20%
Step:    88,    Loss: 1.089,    Acc: 62.97%
Step:    89,    Loss: 1.104,    Acc: 62.60%
Step:    90,    Loss: 1.098,    Acc: 63.17%
Step:    91,    Loss: 1.081,    Acc: 63.27%
Step:    92,    Loss: 1.078,    Acc: 63.64%
Step:    93,    Loss: 1.087,    Acc: 63.17%
Step:    94,    Loss: 1.097,    Acc: 63.30%
Step:    95,    Loss: 1.078,    Acc: 63.44%
Step:    96,    Loss: 1.077,    Acc: 63.30%
Step:    97,    Loss: 1.074,    Acc: 63.34%
Step:    98,    Loss: 1.066,    Acc: 63.47%
Step:    99,    Loss: 1.074,    Acc: 63.77%
Step:  1000,    Loss: 0.681,    Acc: 73.36%
Step:  2000,    Loss: 0.568,    Acc: 77.30%
Step:  3000,    Loss: 0.467,    Acc: 82.68%
Step:  4000,    Loss: 0.436,    Acc: 85.23%
Step:  5000,    Loss: 0.421,    Acc: 85.60%
```

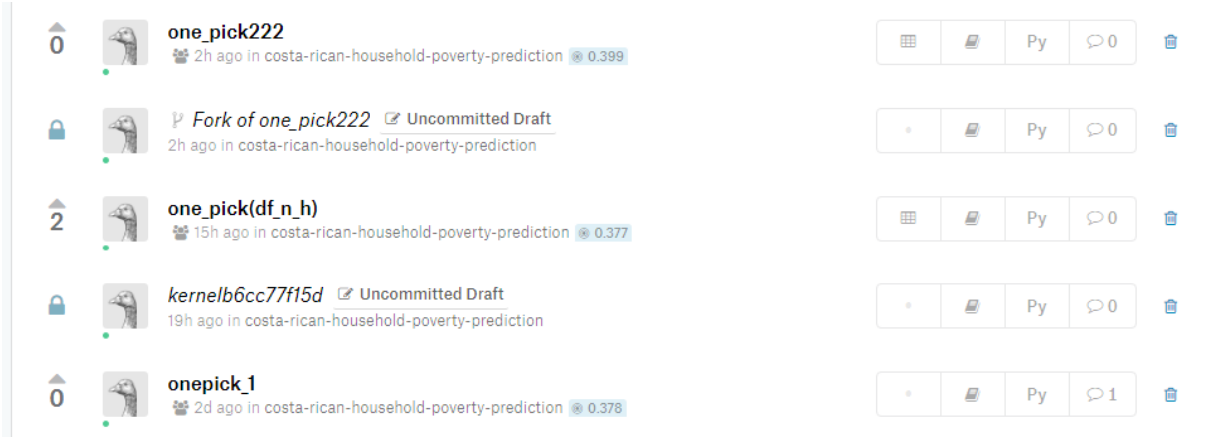
2) *세대주가 아닌 train 데이터에 적용하여 정확도 파악 : 64.61%*

Train 데이터 중 세대주인 경우만 뽑아서 훈련시켰기 때문에 그 외의 세대주가 아닌 사람들의 데이터를 활용해 정확도를 분석할 수 있음

우리의 모델을 통해 예측한 데이터와 실제 결과값을 비교해 정확도를 예측

4) *실제 테스트 데이터에 적용 : 39.9%*

Kaggle에 제출된 결과 중 최고점수가 44.8%인 프로젝트에서 39.9%를 기록할 수 있었다.



5) *한계 및 의의*

- (비취약계층)의 빈도수가 높다는 것은 예측과 동일
- 그러나 극빈계층을 반영하는 피쳐값들의 가중치 고려가 부족해 위와 같이 1의 비율이 매우 저조하게 나온 것으로 예측됨
- 실제로 빈곤함에도 불구하고 위의 모델로는 극빈가구층에 대한 지원이 충분히 이루어지지 않을 수 있다는 위험이 존재하기에 이에 대한 고려가 필요

- 피쳐값에 대한 이해를 기반으로 통계적으로 수치를 가공해 분석하는 과정이 머신러닝에 있어 매우 핵심적이라는 것을 학습

.

감사합니다