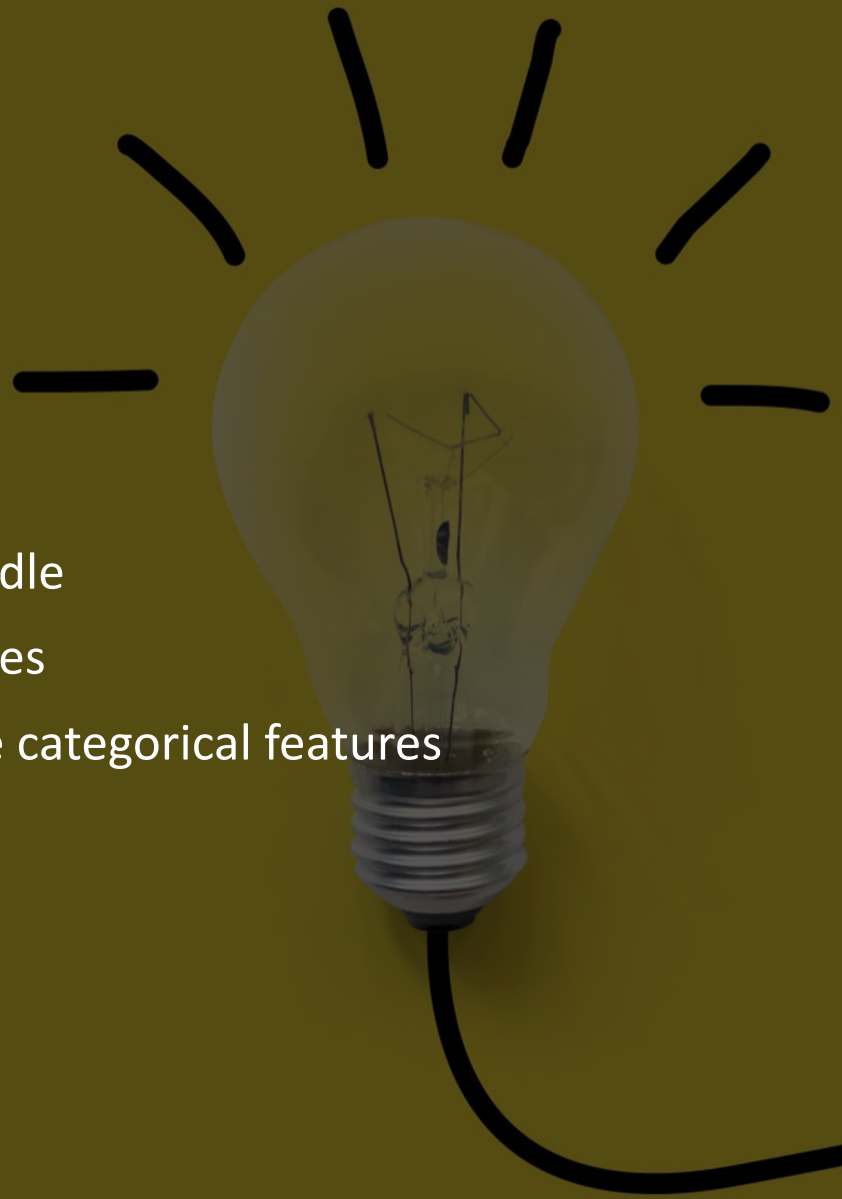
# Click Prediction

How to handle

- Time-series
- Large size categorical features



김푸르뫼 kprm2000@gmail.com

# Contents

- Data exploration
- Data preprocess
- Modeling strategy 1: Spark sparse matrix
- Modeling strategy 2: PCA
- Modeling strategy 3: Categorical embedding

# Modeling Goal

- 웹 기반 광고의 클릭 발생 데이터를 분석하여, 클릭 후 실제 구매 행위로 이어지는 확률 예측

click_timestamp : 광고 클릭 발생시간,

integer_feature_1~8: 정수형 속성 8개

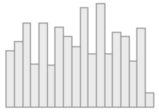
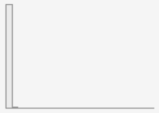
categorical_feature_1~9: 범주형 속성 9개

label: 광고를 클릭한 유저가 구매행위를 수행했는지 여부 (binary)

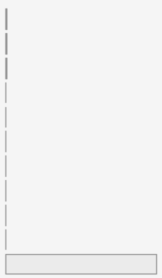
# Data Exploration

- R summarytools library
- 변수별 분포, min, max, median, average values 및 NA ratio 확인
- 8개의 정수형 변수, 9개의 범주형 변수 존재

**Dimensions**: 2536535 x 19
**Duplicates**: 137

| No | Variable | Stats / Values | Freqs (% of Valid) | Graph | Valid | Missing |
|---|---|---|---|---|---|---|
| 1 | click_timestamp [numeric] | Mean (sd) : 437025.5 (243065.8)<br>min < med < max:<br>7 < 454739 < 863999<br>IQR (CV) : 417492 (0.6) | 699728 distinct values |  | 2536535 (100.0%) | 0 (0.0%) |
| 2 | integer_feature_1 [numeric] | Mean (sd) : 1.7 (5.1)<br>min < med < max:<br>0 < 0 < 496<br>IQR (CV) : 1 (3) | 171 distinct values |  | 625177 (24.6%) | 1911358 (75.4%) |

... ...

| | | | | | | |
|---|---|---|---|---|---|---|
| 10 | categorical_feature_1 [character] | 1. 6d295ec5<br>2. b42a9a0f<br>3. 7f6438ab<br>4. a207ca8c<br>5. ad6e0317<br>6. 0680dabb<br>7. 620438e8<br>8. 8a242e67<br>9. fde122fc<br>10. d4a4d6f1<br>[ 37764 others ] | 18484 ( 0.7%)<br>16853 ( 0.7%)<br>14876 ( 0.6%)<br>11582 ( 0.5%)<br>10953 ( 0.4%)<br>9984 ( 0.4%)<br>9983 ( 0.4%)<br>9462 ( 0.4%)<br>9327 ( 0.4%)<br>8120 ( 0.3%)<br>2416911 ( 95.3%) |  | 2536535 (100.0%) | 0 (0.0%) |

| | | | | | | |
|---|---|---|---|---|---|---|
| 19 | label [numeric] | Min : 0<br>Mean : 0.2<br>Max : 1 | 0 : 1974288 ( 77.8%)<br>1 : 562247 ( 22.2%) |  | 2536535 (100.0%) | 0 (0.0%) |

# Data Exploration

- 정수형 변수들은 최솟값 0 또는 -1에 치우친 분포를 지니며, strong outliers로 인한 longtail 구조
- 정수형 변수 1, 4, 6은 30% 이상 결측(NA)돼 있으며 2, 3, 5, 7, 8번은 2%미만 결측
- train.csv 기준 catogorical features는 변수 당 최대 37774개의 항목 보유
- 전체 2536535 레코드 중 137개가 중복
- Label ratio: 77.8%의 레코드가 0, 나머지 22.2%가 1
- Train, validation, test datasets의 click_timestamp는 정렬된 상태

# Data preprocess

- 정수형 변수:
  - 결측 값은 평균이 아닌 중위 값으로 대체(strong outliers 때문)

- 범주형 변수:
  - NaN값을 문자열 'nan'으로 대체
  - 추후 훈련 데이터에 없는 항목이 나올 경우 Out of Vocabulary (OOV)처리
  - One-hot-encoding matrix 변환:한 변수당 최대 2,536,535 x 37,774 크기의 large matrix 발생

# Modeling strategy

1. Spark sparse matrix를 이용해 차원축소 없이 모든 훈련데이터 사용
2. 주성분 분석(PCA)를 이용한 One-hot-encoding  matrix 차원축소
3. 인공신경망의 Categorical embedding layer 이용

# Modeling strategy1: Spark

- Google Cloud Platform(GCP)의 dataproc이용
- Master node 1, Worker nodes 2
- Pyspark OneHotEncoder & LogisticRegression library
- 희소 행렬을 이용해 차원 축소 없이 모델 훈련 가능
- Logistic regression accuracy: 0.7783
- Gradient-Boosted Trees & MultilayerPerceptronClassifier: out of memory (모델 훈련 실패)

# Modeling strategy2: PCA

- 훈련데이터의 일부(20%)를 표본 추출 한 뒤, PCA를 이용해 one-hot-encoding matrix 차원축소
- 항목별 주성분을 key-value pair로 매핑(python dictionary)
- 변환된 validation dataset의 크기는 4.6GB (PCA rank=50기준)
- Validation dataset 기준, 관측되지 않은 항목(OOV)의 비율은 1% 미만
- Logistic regression accuracy: 0.8235 -> better than the first approach

# Modeling strategy2: PCA

- Accuracy향상을 위해 추가적인 feature engineering 진행
- 시계열 데이터의 특성 활용

| | click_timestamp | integer_feature_1 |
|---|---|---|
| 0 | -1.797944 | -0.158484 |
| 1 | -1.797940 | 0.220638 |
| 2 | -1.797907 | -0.158484 |
| 3 | -1.797898 | -0.158484 |
| 4 | -1.797886 | -0.158484 |

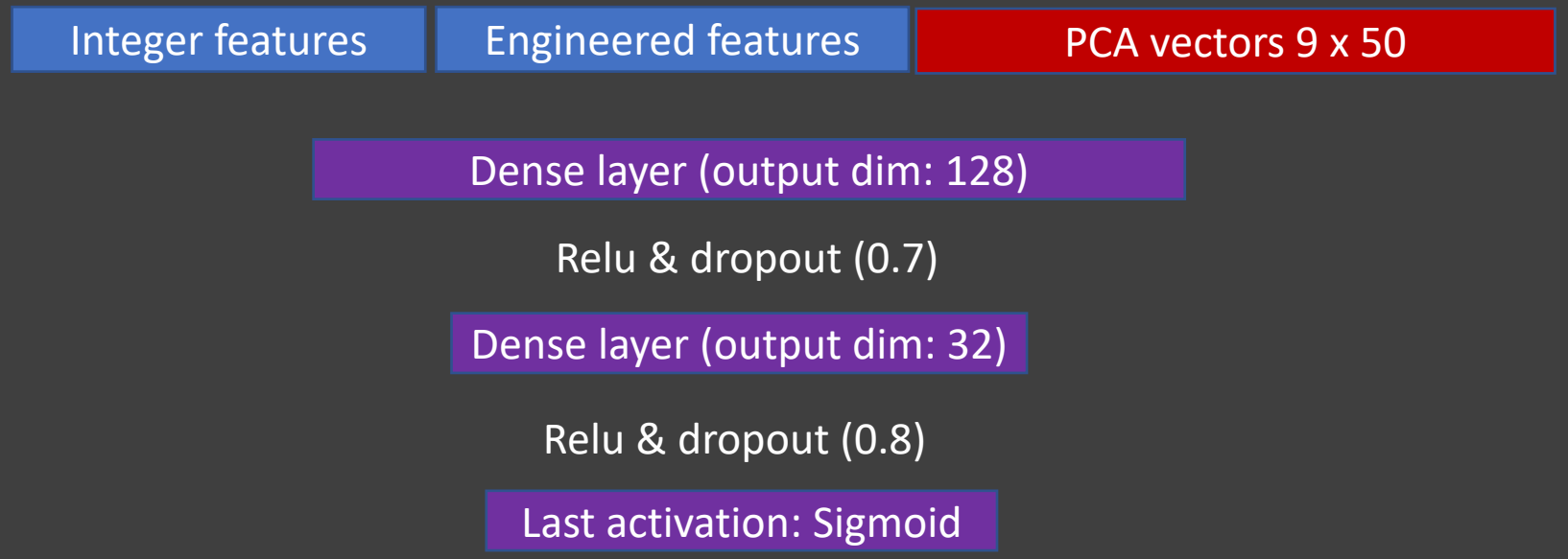
이전과 현재 행을 포함하여 10개 행의 평균/표준편차

이전과 현재 행을 포함하여 5개 행의 평균/표준편차

이전과 현재 행을 포함하여 2개 행의 평균/표준편차

# Modeling strategy2: PCA

- Feature engineering 이후 인공 신경망 훈련

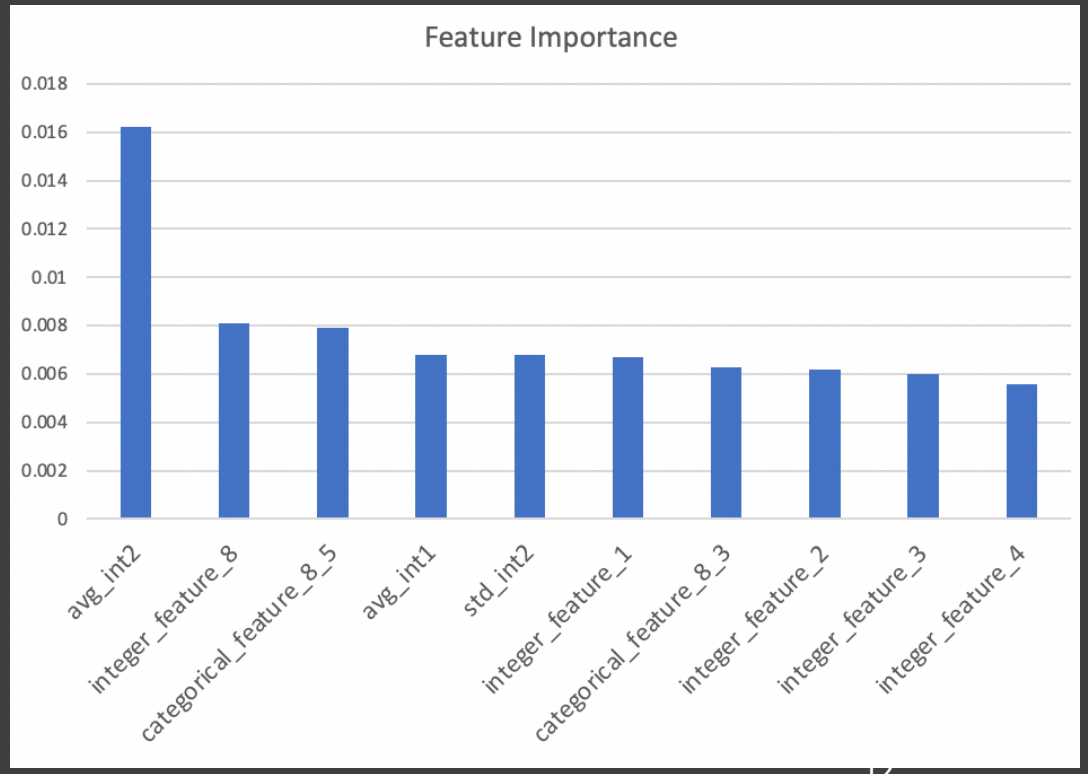



**accuracy: 0.8240**

# Modeling strategy2: Feature importance

- perturbance importance: input variable에 무작위로 노이즈를 더한 뒤, 평균적으로 결과값에 미치는 영향을 계산

| Feature name | Importance | Description |
|---|---|---|
| avg_int2 | 0.0162 | 정수형 변수2의 현재와 바로 직전 시점의 평균값 |
| integer_feature_8 | 0.0081 | 정수형 변수8의 현재 값 |
| categorical_feature_8_5 | 0.0079 | 범주형 변수8의 주성분1 |
| avg_int1 | 0.0068 | 정수형 변수1의 현재와 바로 직전 시점의 평균값 |
| std_int2 | 0.0068 | 정수형 변수2의 현재와 바로 직전 시점의 표준편차 |
| integer_feature_1 | 0.0067 | 정수형 변수 1 |
| categorical_feature_8_3 | 0.0063 | 범주형 변수8의 주성분3 |
| integer_feature_2 | 0.0062 | 정수형 변수 2 |
| integer_feature_3 | 0.006 | 정수형 변수 3 |
| integer_feature_4 | 0.0056 | 정수형 변수4 |

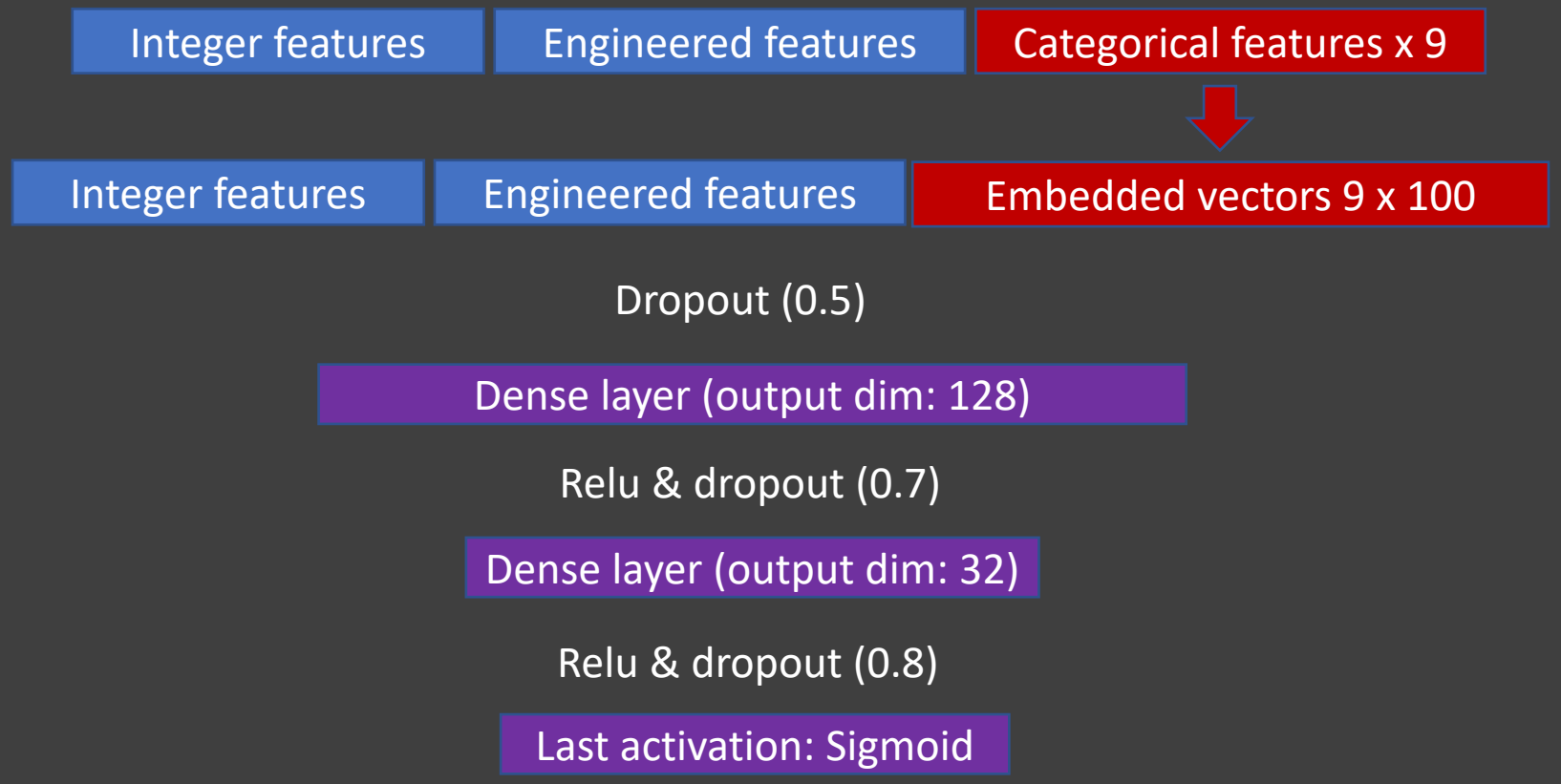


# Modeling strategy2: PCA

- 문제점: 대용량 데이터
  - 메모리 제한으로 디스크에 변환된 데이터셋 저장 필요
  - test, validation 및 표본 추출된 20%의 Train datasets은 각각 4.6GB
  - 디스크 접근으로 인한 속도 저하

# Modeling strategy3: categorical embedding

- 인공신경망에 임베딩 레이어 추가



# Modeling strategy3: categorical embedding

- 추가적인 데이터 처리 불필요
- **Accuracy: 0.8359**
- 최종 모델

# Conclusion

- Spark 이용시 차원 축소 없이 모델링이 가능하나, 기본 GCP  dataproc 보다 더 큰 메모리 자원 필요
- 대형 임베딩 레이어가 없는 PCA기반 모델이 feature importance측정에 편리
- Categorical embedding 모델이  가장 높은 validation accuracy 기록: 0.8359
  - 메모리 및 디스크 또한 효율적으로 이용