아이디어웨어 기업협업 문자 메세지 분류 프로젝트

AI_02_김진우

목차

- 프로젝트 주제
- 프로젝트 진행방향
- 데이터 전처리
- FastText (단어 벡터화)
- DBSCAN (클러스터링)
- 분류 결과 및 분석
- 기대효과 및 한계점
- 프로젝트 회고

프로젝트 주제

※ WISE RETAIL Service 개요

WISE RETAIL

신용카드, 계좌이체, 휴대폰 결제 분석

- 한국인이 많이 결제하는 2,000개+ 소매 브랜드 실시간 결제분석
- 결제금액, 취소금액, 평균결제금액, 구매연령 등 제공
- 한국인 경제활동인구의 6%인 145만명의 표본으로 측정
- 경쟁분석, 전략기획, 마케팅, 투자결정에 활용
- 온/오프라인 소매/서비스 기업, 투자회사 이용 중

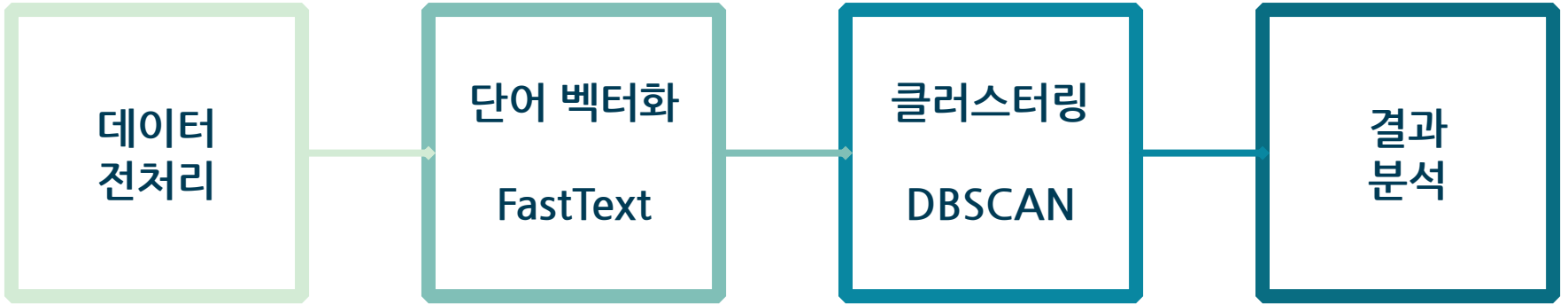
※ 프로젝트 목표 - 유사 단어 분류/클러스터링

기존 : 결제 내역 메세지들을 정규식과 유사한 방식으로 분류 진행

목표 : AI를 활용한 메세지 유사 단어 분류 및 클러스터링

프로젝트 진행방향

※ 프로젝트 진행 방향



데이터 전처리

구 분	내 용
특수문자	특수문자 삭제
숫자/문자 분리	숫자/문자 분리를 통해 숫자만 있는 행 삭제
사람 이름	사람 이름 데이터 셋을 활용하여 이름이 포함되어 있는 행 삭제
빈도수	빈도수 1~3번 메세지 삭제
특정 단어	주식회사, 재단법인, 세금, 출금, 내용없음 등 분류에 무의미한 단어 삭제
한 글자	위 전처리 후 한 글자만 남아 있는 행 삭제

FastText

단어 벡터화(Vectorize)

- 문자 메세지의 단어들을 컴퓨터가 이해할 수 있도록 벡터로 만들어 주는 작업
- 벡터화 방법
  → 등장 횟수 기반 : Bag-of-Words, TF-IDF
  → 분포 기반 : Word2Vec, Glove, FastText
- FastText 사용
  → 메세지의 단어가 비형식적이기 때문에 말뭉치에 포함되지 않은 단어들을 유추가 가능하기 때문

ex)

<eating>

3-grams <ea eat ati tin ing ng>

※ n-gram이 3일 경우 철자 단위 3개로 묶어 유사 단어를 찾을 수 있는 방식

FastText

※ 전처리한 메세지의 유니크 값(중복제거)으로 말뭉치를 만들어 사용

유니크 값이기 때문에 각 단어당 30번 곱하여 말뭉치를 만듦

※ FastText Model 구성

말뭉치 리스트, size = 700, window = 3, min_count = 3, workers = 4, sg = 1, seed=41

```
print(ft.most_similar("nights"))
```

```
[('night', 0.9999917149543762),
 ('rights', 0.9999875426292419),
 ('flights', 0.9999871850013733),
 ('overnight', 0.9999868273735046),
 ('fighters', 0.9999852776527405),
 ('fighting', 0.9999851584434509),
 ('entered', 0.9999849796295166),
 ('fight', 0.999984860420227),
 ('fighter', 0.9999845027923584),
 ('night.', 0.9999843835830688)]
```

※ 결과 예시 (데이터 셋 : lee_background.cor)

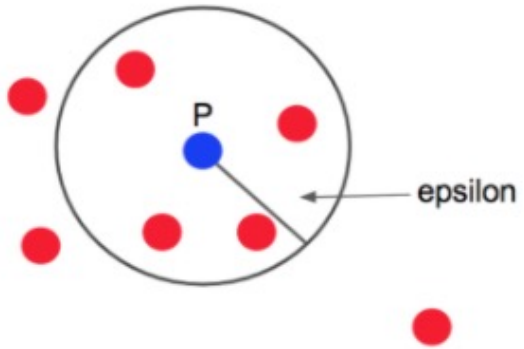
- 말뭉치에 nights라는 단어가 없더라도 말뭉치 내의 유사 단어 night, rights 등을 출력할 수 있음

DBSCAN

※ DBSCAN(밀도 기반 클러스터링)

· 밀도가 높은 부분을 클러스터링 하는 방식 (기준점 반경에 n개 이상이면 하나의 군집)

· K-Means와 차이점은 클러스터 수를 지정하지 않아도 되며 밀도에 따라서 서로 연결하기 때문에 기하학 적인 군집도 찾을 수 있는 것이 큰 장점



※ 하이퍼 파라미터

· eps = 0.01, min_samples = 4

· 기준점 P에서 반경 거리(0.01) 내의 개수(4개)를 지정

· 상기 조건에 맞는 클러스터링 진행

DBSCAN

※ DBSCAN(eps=0.01, min_samples=5, metric = "cosine") → 결과 33% 분류

※ 성능 개선 시도

- 반경 거리내 수(min_samples)를 큰 수에서 작은 수로 반복 분류 작업 실시
- 클러스터링 1회 진행 후 미 분류 데이터만 모아 반복적으로 진행 하는 방식

구 분	minPts 10~2	minPts 100~2
클러스터링 횟수	9	44
분류 건	1,316	1,477
분류 비율 (유니크 값, 건수)	42.2%	45.4%
분류 비율 (전체 데이터 적용, 건수)	44.7%	46.1%

DBSCAN

※ DBSCAN 클러스터링 결과 예시

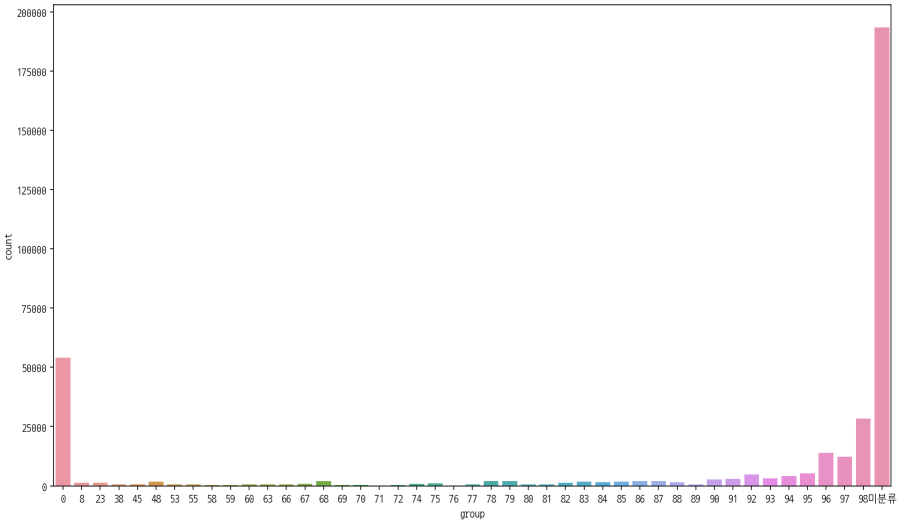
클러스터링 No.	minPts	분류	분류 건수	그룹이름 예시
0	100	1~21	21	0:1
8	92	0	1	8:0
68	32	0~4	5	68:0
87	13	0~13	14	87:0
91	9	0~23	23	91:0
96	4	0~122	123	96:0
98	2	0~695	696	98:0

- DBSCAN 반복 99회
- 총 클러스터는 44회 진행
- 클러스터는 minPts가 작을수록 분류 건수가 증가함
- 분류 그룹 이름 지정
  → “클러스터링No.: 분류No.”

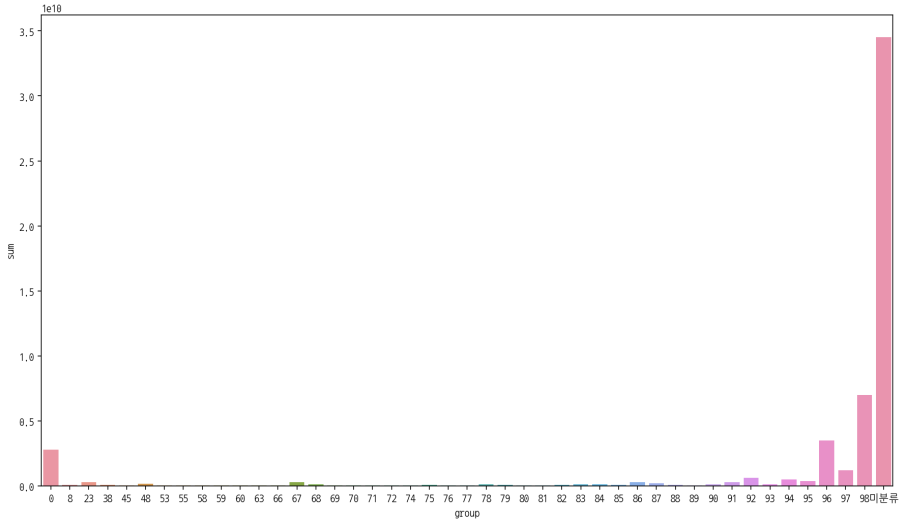
※ 분류 그룹명으로는 어떤 단어가 포함되어 있는지 확인 불가

분류 결과 및 분석

※ 클러스터링 No. 기준 시각화 (전체 데이터 적용, 35만9천건)



〈빈도수〉

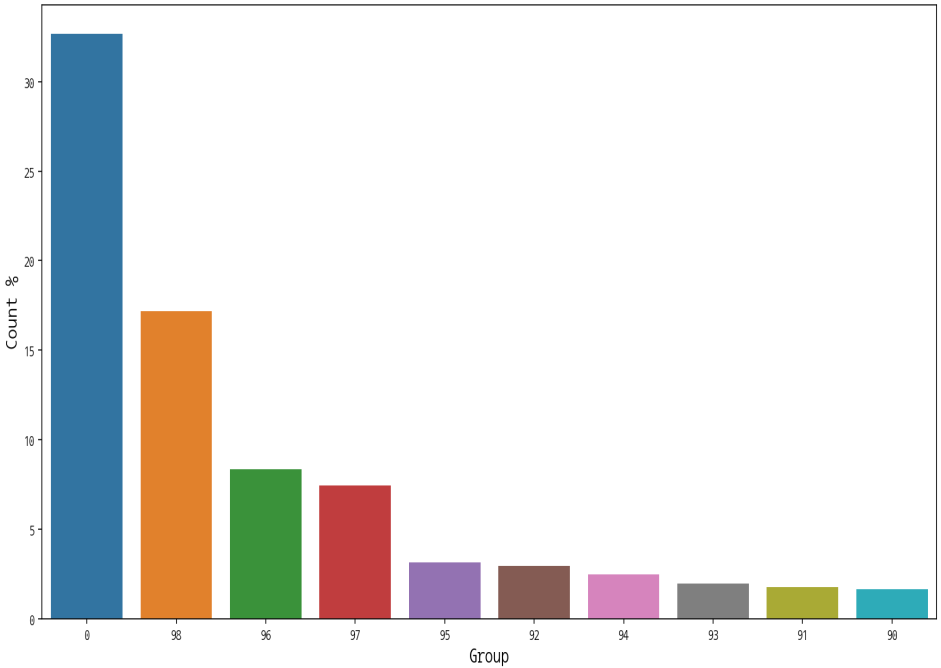


〈금액합〉

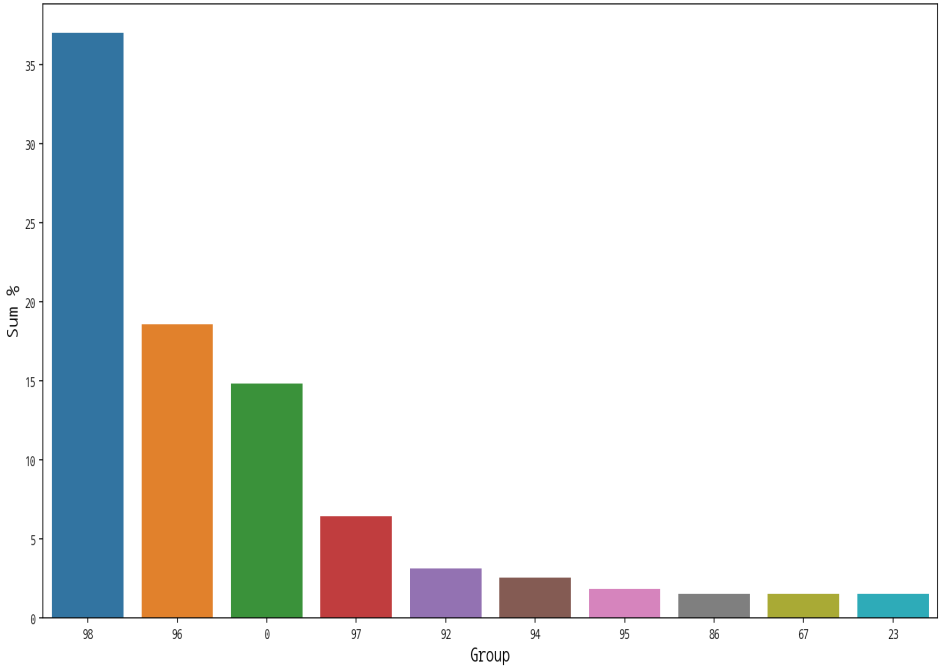
미 분류가 약 54%를 차지하여 분류 상세를 확인하기엔 어려움 → 미 분류 제외 후 분석

분류 결과 및 분석

※ 클러스터링 No. 기준 시각화 - 빈도수, 금액합



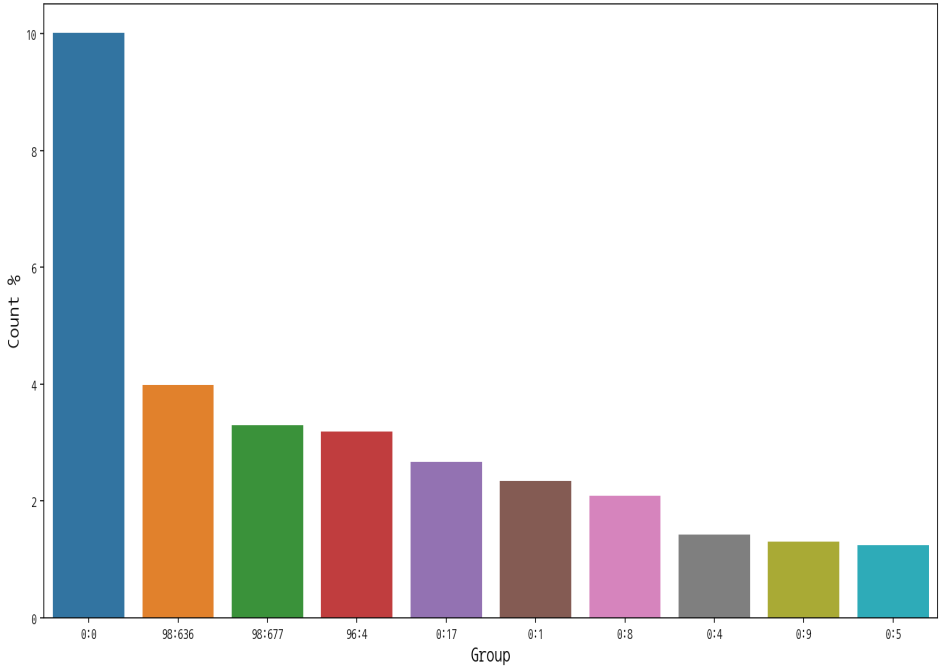
〈빈도수 TOP10〉



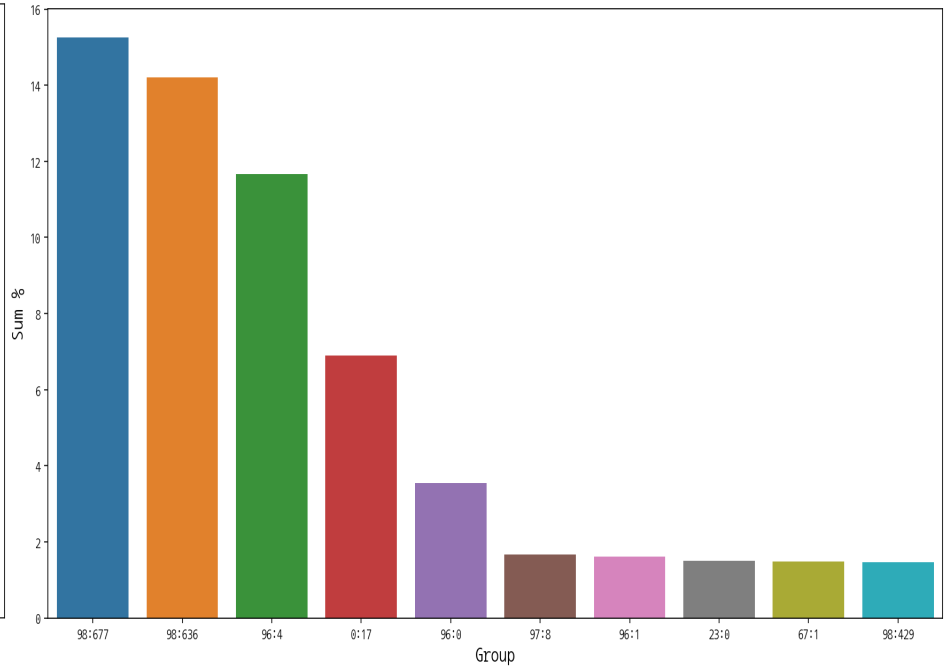
〈금액합 TOP10〉

분류 결과 및 분석

※ 상세 그룹 기준 시각화 - 빈도수, 금액합



〈빈도수 TOP10〉



〈금액합 TOP10〉

프로젝트 기대효과 및 한계점

001 >> 분류 자동화 및 매출 관리

- 분류 시간 단축 및 미 분류 메세지의 분류(클러스터링) 진행 가능
- 분류 된 데이터 관리를 통해 매출 및 빈도를 확인하여 추가적인 업체 관리 가능

002 >> 한계점

- FastText의 경우 문장이 아닌 단어를 강제로 이어 붙인 말뭉치를 사용하였고, 모델 성능을 육안으로 확인할 수 밖에 없음
- 미 분류 메세지 중 여전히 54%는 DBSCAN으로 분류할 수 없음

003 >> 더 시도해 볼 사항

- FastText 및 클러스터링에 대한 논문에 대해 알아볼 것 (비지도 학습)
- 단순한 단어 분류에 대해 추가적인 방법론에 대해 알아볼 것 (지도 학습)

프로젝트 회고

001 >> 힘들었던 점

- 주제와 데이터 셋의 경우 명료하고 간단 했지만, 제한된 데이터로 유사 단어 분류를 해야한 다는 점 (난이도)
- 생각 했던 것 보다 분류 비율이 낮아 성능 향상에 대한 어려움

002 >> 좋았던 점

- 기업의 실제 데이터와 고민에 대해 다루어 본 경험
- 기업 협업 프로젝트가 아니 였으면 경험해 보지 못할 기능들을 다뤄본 것
- 대량의 데이터를 전처리 시 조금이나마 빠르게 하는 법을 알게 된 것
- 동기와의 커뮤니케이션을 통해 자료 공유 및 방법론에 대한 의견을 나눠보고 같은 주제를 함께 해결해 본 것에 대해 만족함