이남걸

**문제 1.**

**알고리즘 : k-modes(k-prototype)**

x1 ~ x8 변수 중 x1(연령)과 x6(연 평균 잔고)을 제외한 다른 변수들이 모두 명목형 변수들이다. 주어진 명목형 변수들은 각 변수들 사이의 관계를 숫자로 나타낼 수 없다. (학력의 경우 저학력 -> 고학력 순으로 나타나 순위형 변수처럼 보이지만 각 level간의 차이를 저학력 = 1, 고학력 = 3 처럼 숫자로 나타낼 경우 ‘저학력 \* 3 = 고학력’이라는 관계가 성립하지 않기 때문에 역시 명목형으로 처리하기로 한다.) 따라서 명목형 변수를 One-hot Encoding 한 수 가장 잘 알려진 k-means 알고리즘을 적용하는 방법을 생각할 수 있겠지만 k-means 알고리즘은 명목형 변수에 적합하지 못하다.

이에 대한 대안으로 등장한 알고리즘 중 k-mode가 있다. 그 중에서도 numerical한 변수와 명목형 변수를 동시에 처리할 수 있는 k-prototype 알고리즘을 클러스터링에 사용하도록 할 것이다. 파이썬에 구현되어 있는 kmodes 패키지를 사용하여 클러스터링을 실시하도록 한다[[1]](#footnote-1).

**Pre-processing**

결측값이 있는 x2와 x4의 경우 명목형 변수이기 때문에 명복형 변수의 결측치를 보정하는 가장 일반적인 방법을 선택하여 최빈값(mode)로 보정하였다. binary 형태인 x5(채무 불이행 유무), x7(주택 자금 대출 유무), x8(개인 신용 대출 유무)은 0, 1의 값을 가질 수 있도록 label encoder를 사용하였다. 한글인 학력은 영어로 대체하였다.

또한 numerical 변수인 x1, x6의 경우 클러스터링 시 동일한scale을 가질 수 있도록 min-max scaler를 사용하여 0과 1 사이의 값으로 변환하였다.

**파라미터(k) 선정 : Elbow Method**

클러스터링을 실행할 때 가장 중요한 것 중의 하나는 클러스터의 수를 결정하는 것이다. 먼저 클러스터를 1~9로 했을 때 cost를 각각 구한 후 그것을 시각화하여 나타나는 elbow로 클러수터 수를 결정할 것이다. k-prototype의 경우 입력 데이터의 숫자가 많아지고 cluster 수가 늘어날수록 계산비용이 증가하여 40,000개의 샘플을 모두 이용하여 이 과정을 진행하는 것은 비효율적이라고 판단하여 10%의 샘플을 random sampling하여 진행하였다.

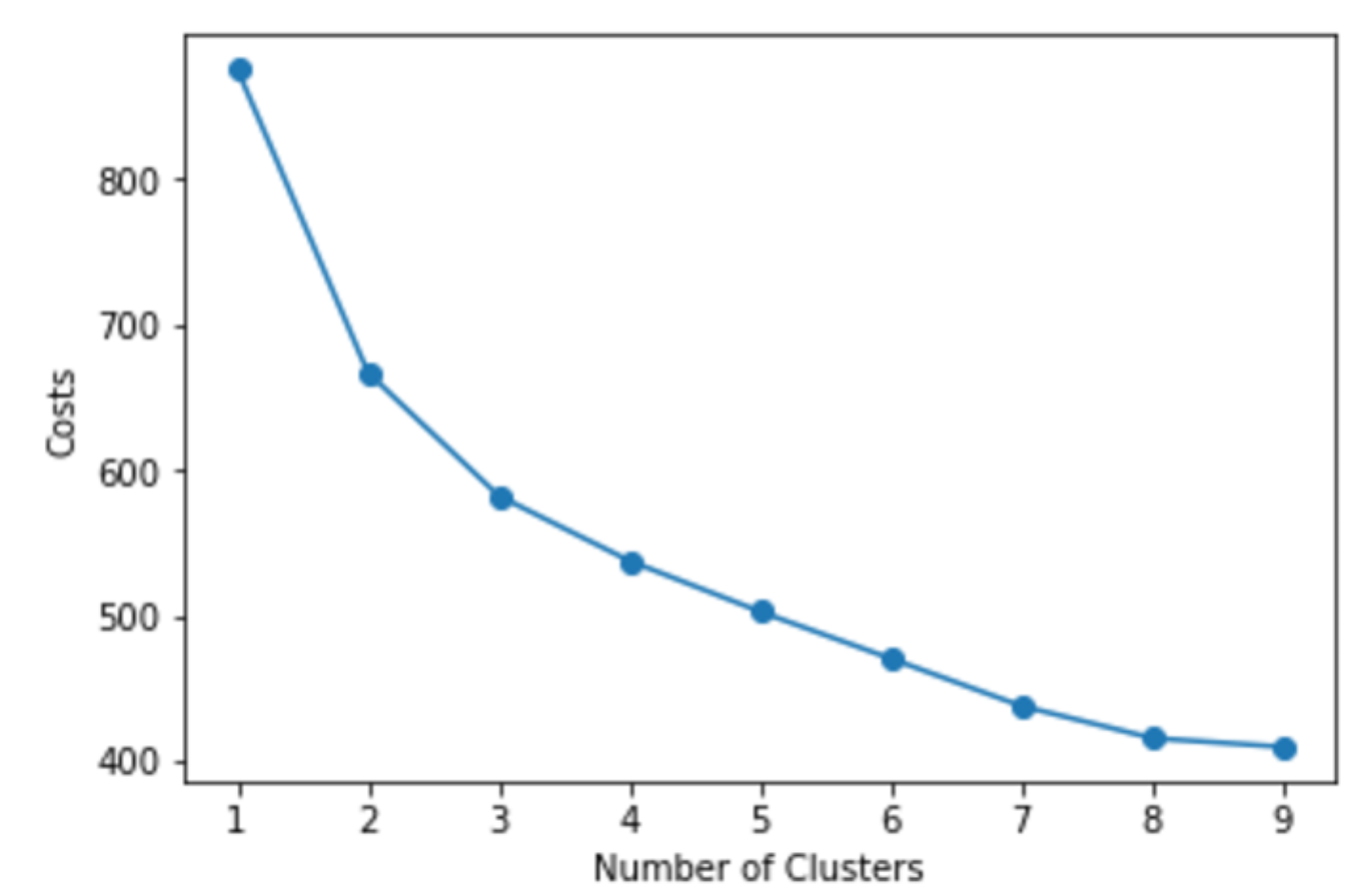


Figure 1. Elbow method plot

위에서 보는 바와 같이 클러스터가 3개가 된 이후 cost는 안정적인 기울기를 가지는 것으로 보인다. 따라서 우리는 elbow point를 3으로 보고 클러스터 수를 3으로 정하여 전체 샘플에 대한 클러스터링을 수행하도록 한다.

**클러스터링 결과**

우선 3개 그룹의 각 centroid로 특징을 분석해 보자.

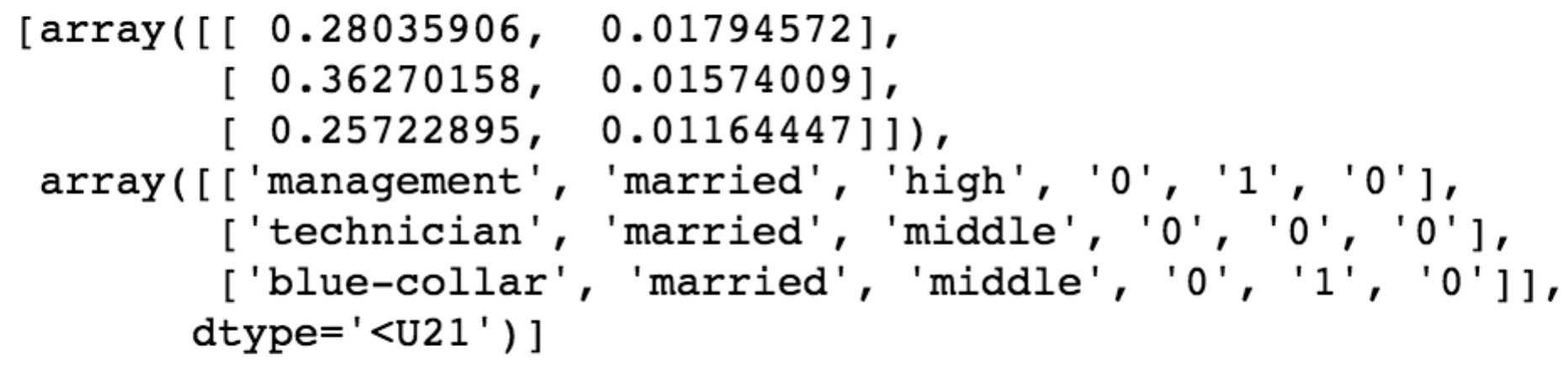


Figure 2. Centroid of each group

위 결과에서 볼 수 있듯이 각 그룹은 나이, 연 평균 잔고, 직업, 학력, 주택 자금 대출 여부에서 차이가 난다. 이 결과를 각 변수별로 좀 더 자세히 살펴 보도록 하자.

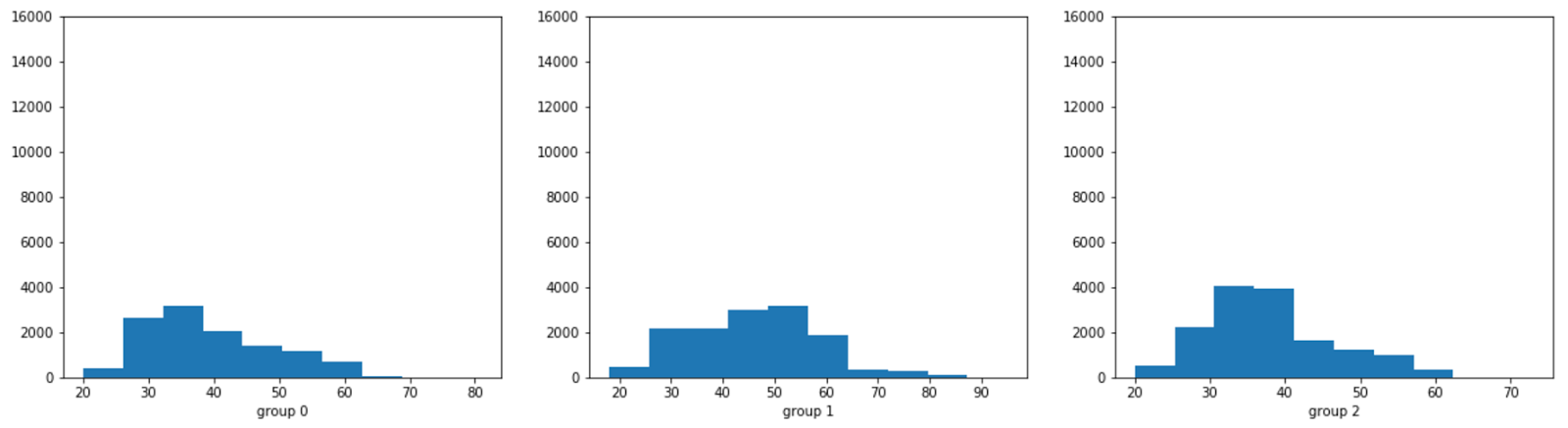


Figure 3. Distribution of Age

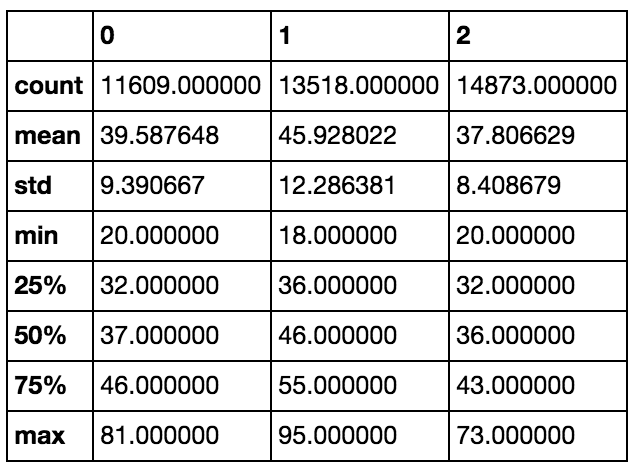


Table 1. Summary of age per group

결과에서 볼 수 있듯이 1번 그룹의 연령이 다른 그룹에 피해 높은 경향을 보이고 있다.

또 다른 numerical 변수인 x6(연 평균 잔고)에 대해 살펴보도록 하자.

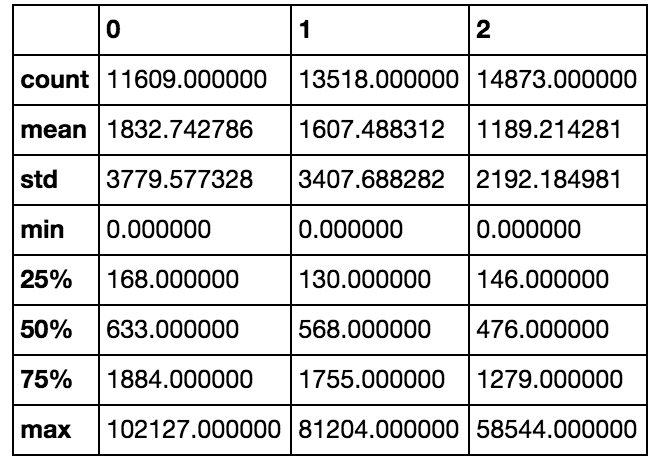


Table 2. Summary of balance per group

연 평균 잔고 변수에서는 0번 그룹이 다른 그룹들보다 높은 경향을 보이고 있다.

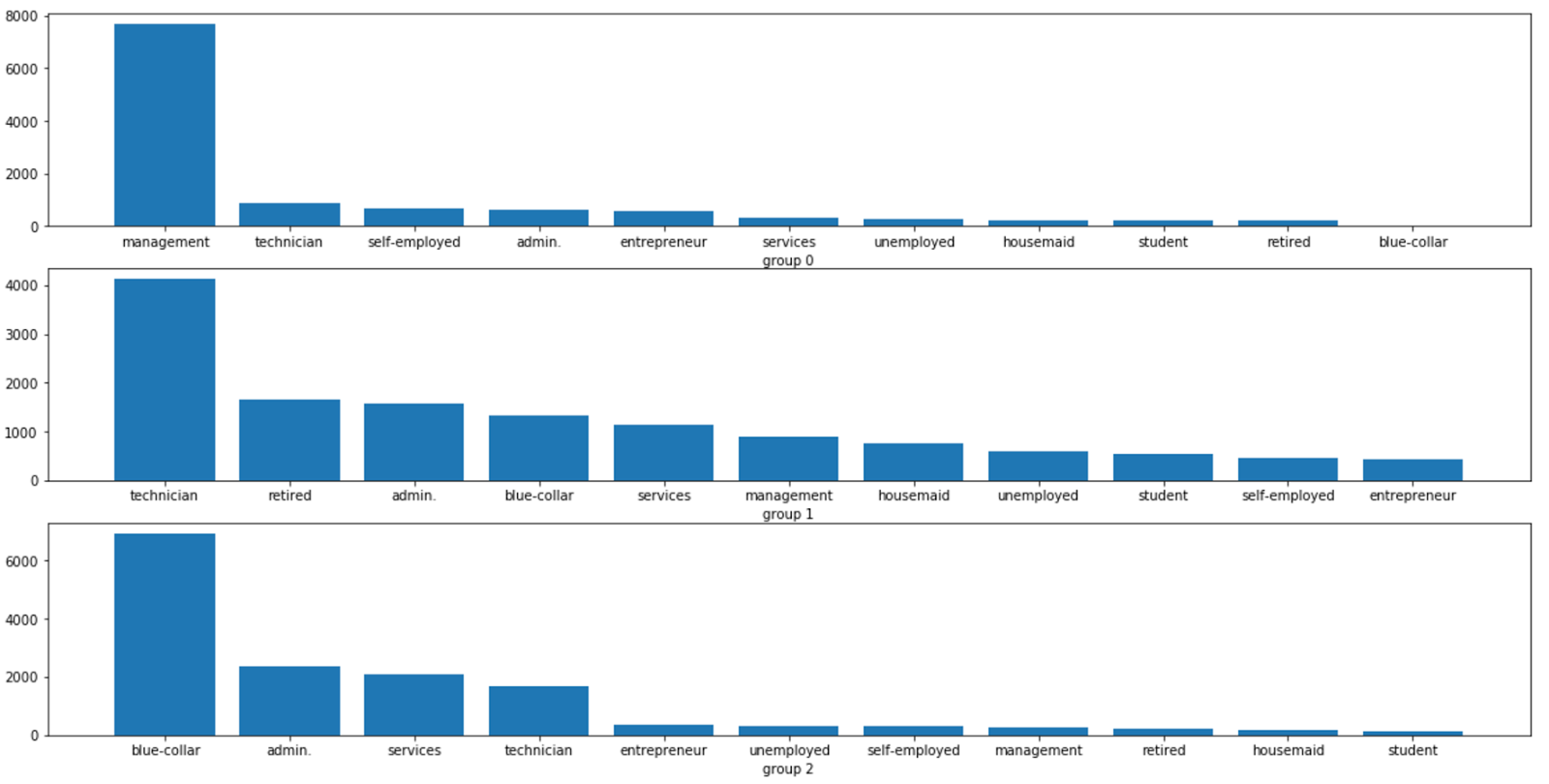


Figure 4. Distribution of Jobs

직업 종류를 살펴보면 0번 그룹에서는 ‘management’가 1번에서는 ‘technician’이 2번에서는 ‘blue-collar’가 많은 경향을 보이고 있다.

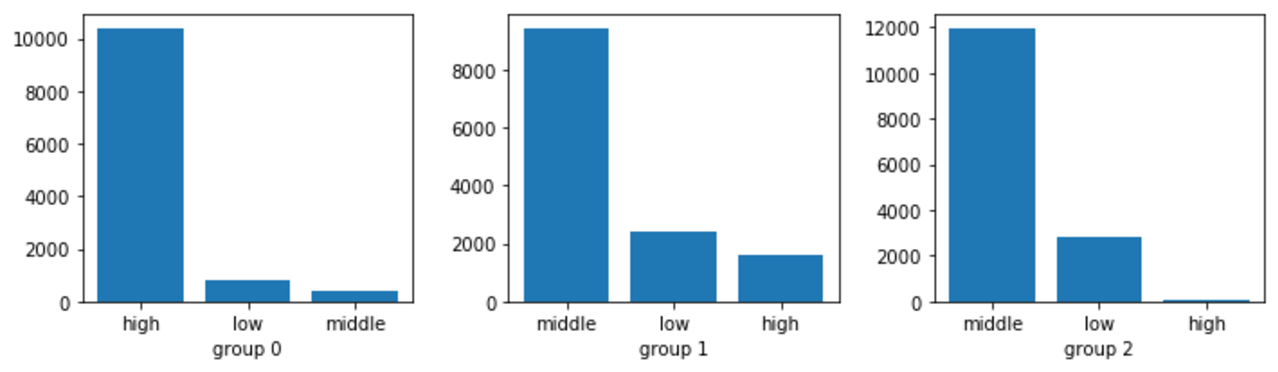


Figure 5. Distribution of Education

교육 수준은 0번 그룹에서는 ‘고학력’, 나머지 그룹에서는 ‘중학력’이 많은 부분을 차지하고 있다.

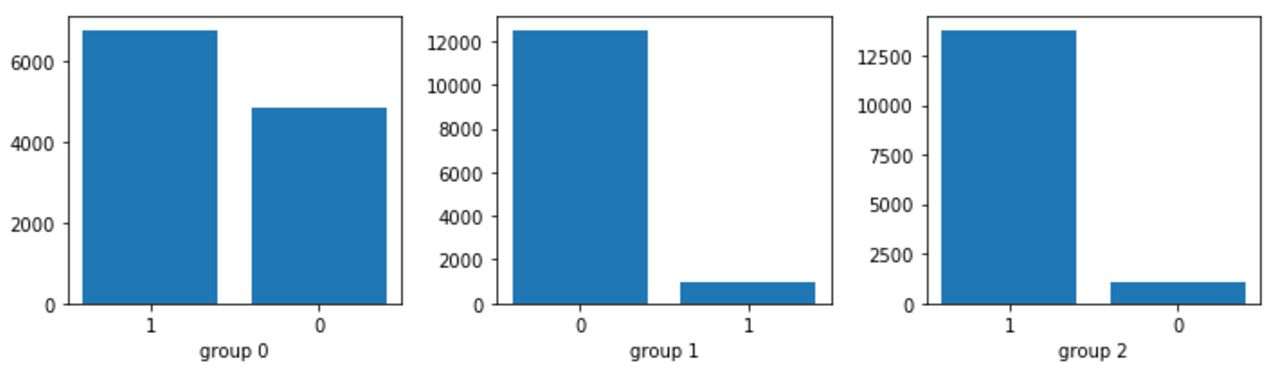


Figure 6. Distribution of Housing Loan

주택 자금 대출 여부는 그룹1과는 달리 그룹 0과 2에서 ‘yes’가 많이 관찰되고 있다.

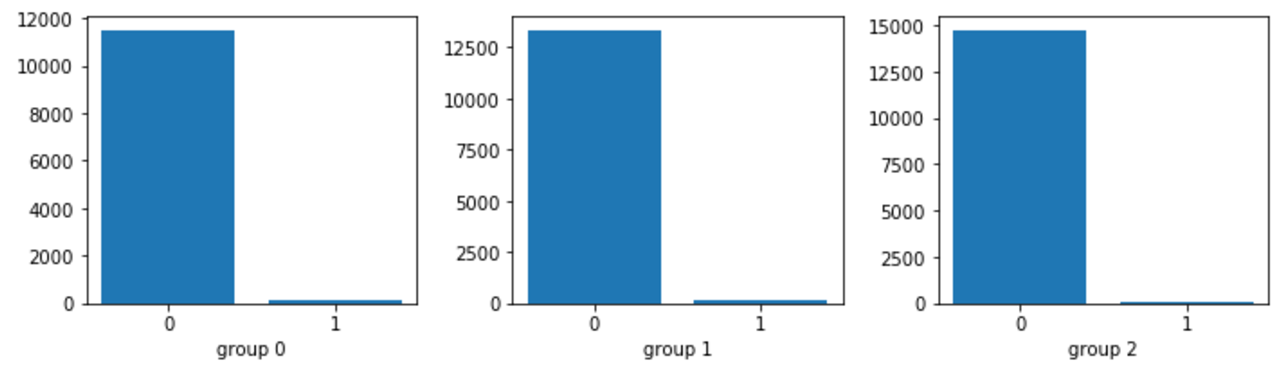


Figure 7. Distribution of Default

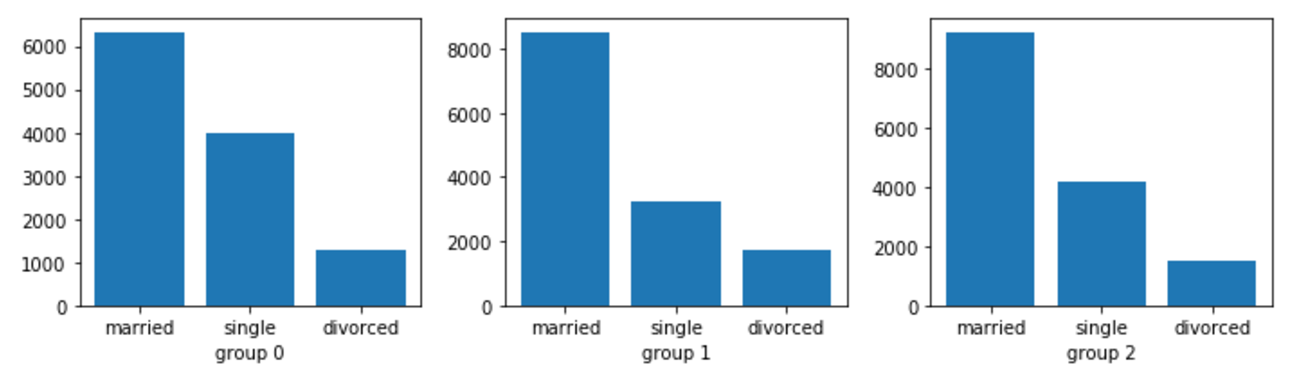
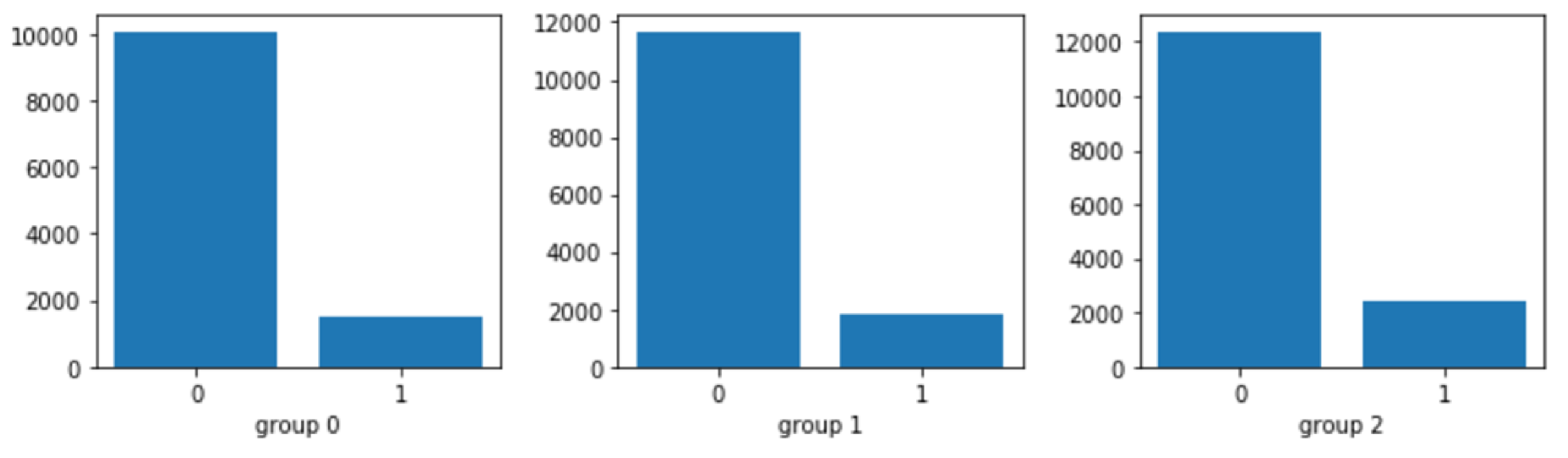


Figure 8. Distribution of Marriage

 Figure 9. Distribution of Credit Loan

위의 다른 변수들과는 달리 ‘채무 불이행 여부’, ‘결혼 여부’나 ‘개인 신용 대출 여부’는 그룹 간 뚜렷한 분포 차이를 보이지 않고 있다.

요약을 하면 그룹0은 나이는 조금 어리지만 고학력에 주로 관리자 업무를 하고 있으며 연평균 잔고가 많고 주택 구입을 위해 대출을 많이 한 그룹이라고 할 수 있다. 그룹1은 조금은 나이가 많으며 중간 정도 학력에 주로 기술직(technician)에 종사하고 있고 연평균 잔고가 조금은 적은 그룹이다. 이 그룹에서는 주택 자금 대출이 현저하게 적다. 그룹 2는 나이는 그룹0과 1의 중간이지만 주로 연평균 잔고가 적은 ‘blue collar’ 고객이 많이 속해 있다. 이들의 학력은 ‘중학력’과 ‘저학력’이 대부분이며 ‘고학력’은 거의 없는 것을 알 수 있다. 또한 주택 자금 대출도 다른 그룹보다 높은 양샹을 보인다.

위 클러스터링을 통해 살펴 본 결과 각각의 그룹이 어느 정도 특징을 가지는 것을 볼 수 있다. 이런한 고객 cluster를 활용하여 각종 금융 상품 마케팅을 실시나 고객 관리를 할 수 있을 것이다.

**문제 2.**

**Pre-processing**

결측값이 있는 x2와 x4, x9는 문제1과 마찬가지로 최빈값(mode)로 보정하였다x13(이전 캠페인에서 마지막으로 고객과 연락된 후로부터 경과된 일수)과 x15(이전 캠페인의 결과)는 결측값이 32,484건이나 되어 다른 방법으로 보정하였다. x13의 경우 이전 캠페인에 연락이 된 경우 1, 아닌 경우 0이 될 수 있게 변환하였다. 또한 x15는 이전 캠페인에 성공한 경우 1, 아닌 경우 0이 되게 하였다.

binary 형태인 x5(채무 불이행 유무), x7(주택 자금 대출 유무), x8(개인 신용 대출 유무)도 마찬가지로 0, 1의 값을 가질 수 있도록 label encoder를 사용하였다. numerical 변수인 x1, x6, x12, x14의 경우 동일한scale을 가질 수 있도록 min-max scaler를 사용하여 0과 1 사이의 값으로 변환하였다.

나머지 명목변수(x2, x3, x4, x9, x10, x11)은 one-hot encoding을 실시하여 변환하였다. y값의 경우도 0, 1의 값을 가질 수 있도록 label encoder를 적용하였다.

**Split Train and Test Data**

훈련데이터와 테스트 데이터를 80%와 20%로 분류하였다. 모델들을 성능을 평가할 때는 테스트 데이터 내에서 Fold가 10인 Strata Cross Validation을 사용하였으며 20%의 테스트 데이터는 훈련 과정에 참여할 수 없게 하여 마지막에 모델의 최종 성능을 측정하는데 사용하였다.

**Feature Selection**

피쳐가 너무 많은 경우 오버피팅의 원인이 될 수 있기 때문에 피쳐의 수를 줄이기 위해 feature selection을 실시하였다. 하나의 방법은 random forest의 결과에 피쳐 중요도를 파악하여 선택하는 것이고 또 다른 방법은 Logistic Regression 모델로 부터RFE(Recursive feature elimination)를 실시하는 것이었다.

가장 널리 쓰이는 분류기인 Random Forest와 Logistic Regression으로 테스트 한 결과 random forest로 부터 23개의 변수 (x1, ,5, x6, …, x3\_single)을 선택하였을 때와 RFE로 2개의 변수(x12, x15)를 선택하였을 경우의 accuracy결과가 RFE 쪽이 약간 우세하고 사용하여야 할 변수를 줄이는 것이 좋다는 결론 하에 x12, x15를 변수로 선택하였다.

**Model Selection & Evaluation**

사용할 모델을 선택하기 위해 가장 널리 사용되는 분류기들을 비교하여 보았다.

1) Logistic Regression

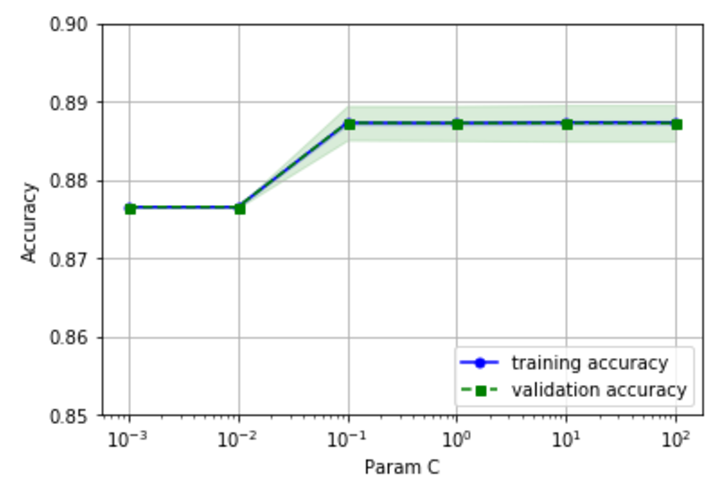


Figure 10. Validation Curve of Logistic Regression

로지스틱 회귀에서 오버피팅 방지를 위해 L2 정규화를 사용할 경우, 정규화 변수의 역수인 C값을 정해주는 것은 중요한 일이디다. 위 그래프는 C값을 정하기 위한 검증 곡선이다. 그래프를 보면 오버피팅이 거의 일어나지 않고 있고 최적의 C값은 0.1부근에서 결정되는 것을 볼 수 있다.

실제 C값을 이용하여 이용하여 훈련 데이터에 대해 10번의 층화 교차 검증을 사용한 결과 약 88.72%의 정확도를 가지는 것으로 나타났다.

2) Random Forest

1000개의 tree를 생성하여 random forest 알고리즘을 실시한 결과 88.77%의 정확도를 보였다. Random forest는 의사결정나무에서 발생할 수 있는 오버피팅이 줄어들 수 있다는 장점을 가지고 있다.

3) SVM

SVM의 경우 linear 커널과 RBF 커널 을 이용했을 경우, 정확도가 약 88.27%로 나타났다.

4) MLP

MLP의 경우 hidden layer size를 (5, 2)로 설정하였을 때 약 88.22%의 정확도를 나타냈다.

5) Ensemble – Majority Vote

앙상블 기법 중 다수결 투표를 이용하여 성능 개선을 시도하여 보았다. 로지스틱 회귀, MLP, Linear SVM을 결합하여 다수결 투표기를 만들어서 성능을 평가하여 보았다. 그 결과는 88.27%로 계산 비용에 비해 개선 효과가 크지 않음을 알 수 있었다.

위 모든 과정을 거쳐 성능이 가장 좋았던 Random Forest를 선정하여 지금까지 사용되지 않는 테스트 데이터를 평가하여 보았다. 모델을 훈련할 때 사용하지 않은 테스트 데이터에 대해 89.175%의 성능을 보여 주고 있다.

1. https://github.com/nicodv/kmodes [↑](#footnote-ref-1)