Problem 1

1. 예측 모델을 만드는 과정

str함수를 사용해서 각 변수에 값들이 제대로 들어가 있는지, 타입이 제대로 들어가 있는지 확인 후, unknown이라는 결측 치를 먼저 파악했다. Job, marital, education, default, housing, loan 변수에서 unknown값을 detect할 수 있었다. 이 값들은 일단 NA로 바꿨다. 이들은 모두 factor형 이기 NA값은 각 열 마다 가장 많이 관측된 값으로 넣었다. Job 에서는 admin, marital에서는 married, education에서는 university.degree, default에서는 no, housing에서는 yes, loan에서는 no 값으로 NA를 대체하였다. 예측 모델은 0~1 사이의 결과 값이 나오기 때문에 확률 값으로 해석할 수 있다는 장점을 가진 로지스틱 회귀를 선택하였다. 모델을 만들 때, 독립 변수는 주어진 모든 변수를 사용하였다. 변수가 너무 많아 줄여보려 했으나, 따로 테스트 셋을 만들어 평가해 본 결과 모든 변수를 쓸 때 성능이 가장 높았다.(r코드에는 다양한 시도는 삭제했음)

또, 나이브 베이즈도 시도해 보았는데 그렇게 성능이 뛰어나지 않았다.

1. 결과

따로 validation 데이터 셋을 만들어 본 결과 트레이닝과 테스트셋 모두 에서 AUC가 약0.93 정도가 나왔다.

Problem 2

1. 예측 모델을 만드는 과정

탐색 결과 결측 치도 없었고 이상한 값도 없는 것으로 판단하였다.

수치 예측을 하는 decision tree, 즉 regression tree의 앙상블인 Random Forest모델을 사용하였다. 다수의 regression tree를 조합해 나온 결과의 평균을 예측 값으로 사용한다. 따라서 단일 모델보다 오버 피팅을 피할 가능성이 높고 또 더 좋은 예측력을 가질 수 있다고 생각하였다. 가장 중요한 하이퍼 파라미터인 mtry(각 split에서 고려할 변수 수)와 ntree(랜덤 포레스트에서의 나무 수)는 grid search 를 이용하여 찾았다. (mtry = 30, ntree =2000)

1. 결과

따로 validation 데이터 셋을 만들어 본 결과 트레이닝 셋 에서는 R^2가 0.86으로 높았으나, validation 셋에서는 0.39 정도가 나왔다. 오버 피팅이 일어난 것이다.

하지만 linear regression 과 비교해 보았을 때 상대적으로 높은 결과이다.

Linear regression은 트레이닝과 validation에서 각각 0.31, 0.20의 R^2값이 나왔다.