1. Outer join 으로 합치니까 데이터 수가 너무 많아져서 코랩으로 돌릴 때 자꾸 램 초과로 초기화 됨
   데이터 수가 loan_result.csv : 약6995000 << user_spec.csv: 약15600000 이므로,
   loan_result의 application_id 기준으로 left_join 한 데이터 셋과 left_join에 못 들어간 user_spec 데이터를 분리해서 사용하면 좋을 것 같음.
   (합쳐진 데이터는 대출 여부를 비교하면서 분석, 합쳐지지 못한 user_spec데이터는 대출 근처에도 가지 않은(?) 사람들이라 생각하고 따로 분석)
   ➔ 여기서 의문 점: application_id 가 신청서 번호 라고 되어있던데 is_applied가 대출 신청 여부고... 그럼 application_id가 말하는 신청은 뭔 신청 말하는거지??

2. User_spec과 Loan_result을 application_id 기준으로 outer_join 시킨 후, log_data의 user_id기준으로 left_join 시켜서 log_data 중심의 모두 합친 데이터를 만든 후, 행동횟수와 대출 여부의 관계를 파악하고 어느 행동 횟수가 대출여부에 영향을 많이 끼치는지 파악 예정
   ➔ 성수님이 주신 데이터로 해도 램초과로 안합쳐짐……

3. 시각화는 pandas_profiling으로 해놓고 캡쳐 쭉 해놔서 언제든 참고 가능하게 끔,,
   ➔ 7시간 돌아가다 런타임 끊겨버려서 실패

4. 대출 목적에 따라 묶어서 따로 보는 건 어떨까..! 대출 목적에 따라 대출 규모나 특성이 많이 달라질 것 같음. 작년 혜진님 제주eco에서 지역별로 나눠서 비교한거 같은 비슷한 맥락으로.

5. 대출 연체 할 고객일 시 '핀다' 입장에서는 어떤 조치를 취하는게 더 이득일까? 만약 대출 연체 가능성이 있는 고객도 최대한 다 대출 신청하게끔 유도하는게 이득인지, 한도 등의 재검토에 들어가는게 이득인지 배경지식 부족으로 잘 모르겠지만, 만약 따로 관리하는게 이득이라면, 대출 연체 가능성이 높은 고객을 따로 칼럼을 만들어 분석에 적용하는 것도 좋을 것 같음 (user_spec 데이터와 관련 논문을 통해)

6. Open API(https://data.seoul.go.kr/dataList/OA-21098/A/1/datasetView.do)로 서울시 은행 별 대출금리 정보 데이터가 있는데 '핀다'의 승인 금리랑 비교하는 형태로 활용 가능할 수도 있을 것 같음

7. log_data로 고객 이탈률이 높은 페이지 분석 (https://steadiness-193.tistory.com/167)