<1>  
딥러닝 최종 프로젝트로, 타이타닉호 생존 예측 프로그램을 진행하였습니다.

<2>  
우선 프로그램의 목표, Data랑 모델을 설명하고 각 단계별 예측 결과에 대해 설명하고 결론을 말씀드리겠습니다.

<3>  
시스템 목표로는 탑승자에 대한 나이, 성별, 가족관계, 실제 생존여부의 데이터를 이용하여  
싸이킷런에서 제공하는 나이브 베이즈 분류기법 모델을 사용하여 정확도와 loss율을 구하고  
다양한 시도를 통해 보다 높은 정확도와 낮은 loss를 구하는 것이 목표입니다.

Data로는 탑승자에 대한 정보가 포함된 open data set을 가져왔습니다.  
기존 데이터는 문자열, ? 과 같은 처리하기 힘든 데이터를 가지고 있기에 처리에 용이하도록 0, 1 과 같은 숫자와 nan과 같은 데이터로 변환하고 데이터뷰어를 통해 확인하였습니다.

학습모델의 경우 data set의 80%를 학습 데이터, 20%를 검증 데이터로 실시하였고  
각 데이터의 종류에 따른 편차로 인해 데이터 정규화를 실행하였습니다.

싸이킷런 라이브러리인 나이브 베이즈 분류 알고리즘을 사용하여 간단한 모델을 구축하였습니다.

<4>  
우선 첫 번째 예측입니다.  
데이터의 80%를 바로 학습시켜보았더니 74%의 정확성이 나왔습니다.

<5>  
그 다음 바로 신경망의 층을 증가시켜보았습니다.  
단순히 총 3단의 구조이지만

정확성이 약 79%까지 증가하였습니다.  
단을 조금 증가시켰을 뿐인데, 기존 결과보다 5%가 향상되었습니다.

<6>  
이번엔 동일한 코드에 배치사이즈와 에폭수를 변화시켜보았습니다.  
여기 보이시는 표와 같이 실행시켜 보았는데요, 배치사이즈를 64, 에폭수를 500으로 하였을 때가 가장 높은 효율의 정확성인 82%가 나왔습니다.  
하지만 이번 테스트를 통해 문제점을 발견하였는데요, 같은 배치사이즈와 에폭수로 테스트를 해도 정확도가 약간씩 다르게 나오는 것을 확인하였습니다.

<7>  
다음은 데이터셋의 트레이닝과 테스팅 비율을 변화시켜보았습니다. Test\_size를 0.1에서 0.6  
즉, 테스트 데이터셋의 비율을 10%에서 60%까지 증가시켜보았는데요.  
이에 대한 결과로는 50%까지는 학습이 잘 되지만 절반이 넘는 양을 테스트 데이터셋으로 사용하게 되면 학습이 잘 되지 않는 것을 확인하였습니다. 최대 어큐러시는 80%가 나왔습니다.

<8>  
이번에는 Layer의 수를 증가시켜보았습니다. 기존 3단에서 최대 8단까지 증가시켜보았는데요.  
4단부터 약 4%가 증가하는 좋은 효율을 보였지만 6단까지는 별 차이가 없고, 7단부터는 61%의 정확성으로 효율이 오히려 떨어지는 것을 확인하였습니다.  
이에 대해 Weight 값을 Xavier/He initialization을 사용하여 적절하게 초기화 후 실행시키고 싶었지만 제공된 라이브러리를 사용하여 W 값을 변경하지 못했습니다.

<9>  
마지막으로 텐서보드를 통해 Cost와 Accuracy를 확인을 해보았는데요.  
보시다시피 학습횟수가 증가할수록 이상적인 결과를 도출함을 알 수 있습니다.

그래서 직접 에폭수를 1만까지 증가해서 학습시켜보았는데 결과는 기존 학습결과와 큰 차이가 없었습니다. 이에 대한 원인은 아직 발견하지 못해 더 공부해봐야 할 것 같습니다.  
마지막으로 프로젝트를 진행하면서 다소 아쉬웠던 점이 있다면, W 값을 초기화해서 학습시켜보고 싶었는데 W가 라이브러리 안에 숨어있어서 방법을 몰라서 직접 해보진 못하였는데, 다음에 다시 하게 된다면 직접 모델을 구현해봄으로써 W 값을 포함한 여러가지 값들을 변경시켜보고 다양한 network 모델을 구현해보고 싶습니다.