# 인공지능 과제 리포트

과제 제목: KNN and Logistic Regression with MNIST Data

학번: B911197
이름: 최준영

## 1. 과제 개요

28x28해상도를 가지는 이미지는 0~9사의 수를 나타낸다. 해당 이미지 데이터를 사용하여 KNN모델을 만들고 테스트한다. 테스트를 진행하며 최적의 k(인접 이웃의 수)값을 찾는다.

## 2. 구현 환경

MAC OS에서 Pycharm IDE를 사용하여 구현하였다.

## 3. 알고리즘에 대한 설명

28x28픽셀의 이미지를 1차원 배열로 변환하여 총 784개의 feature를 가진 데이터 포인트들로 변환한다. 그 후 유클리트 거리계산법을 사용하여 주어진 테스트 데이터와 트레인 데이터의 거리를 구하고 거리정보를 인접리스트에 저장한다. 인접리스트에서 k개의 이웃을 추출한다. 그 후 가중치 알고리즘을 사용하여 추출한 이웃들중 선택된 라벨을 테스트 데이터의 라벨과 비교하여 정확도를 확인한다.

## 4. 데이터에 대한 설명

### 4.1 Input Feature

x_train : 28x28이미지를 784길이의 unsigned integer array로 변환한 결과물이다.
y_train : 훈련 셋의 라벨을 가지고 있는 배열이다.
x_test : x_train과 같은 유형의 정보를 가지고 있는 테스트 데이터셋이다.
y_test : 테스트 데이터셋의 라벨이다.

### 4.2 Target Output

클래스는 총 0~9이다.
KNN모델의 run매서드를 실행하면 1, 3, 5, 7, 9로 k값을 변경하고 정확도가 가장 높은 k값을 출력한다.
k값과 정확도에 대한 그래프 역시 확인할 수 있다.

## 5. 소스코드에 대한 설명

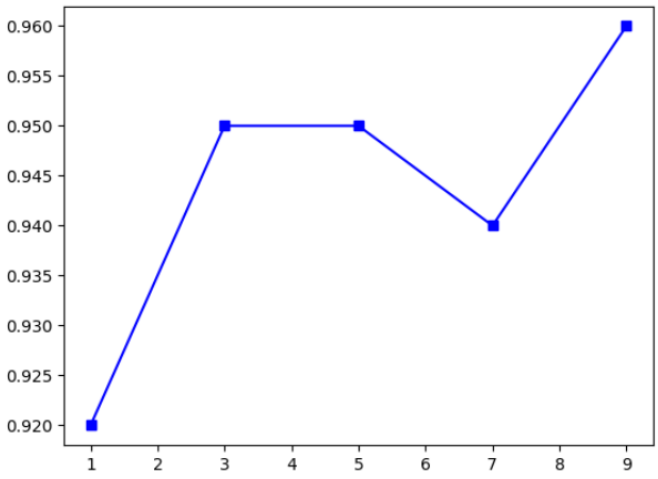
KNN모델을 생성할때 테스트 데이터와 트레인 데이터를 전달한다.
과제에 포함된 전체 데이터는 60,000개 이지만 테스트시간상의 이유로 학습데이터는 10,000개 테스트 데이터는 100개로 제한하였다.
run호출을 통해 적합한 k값을 찾는 정확도 연산을 시작한다.

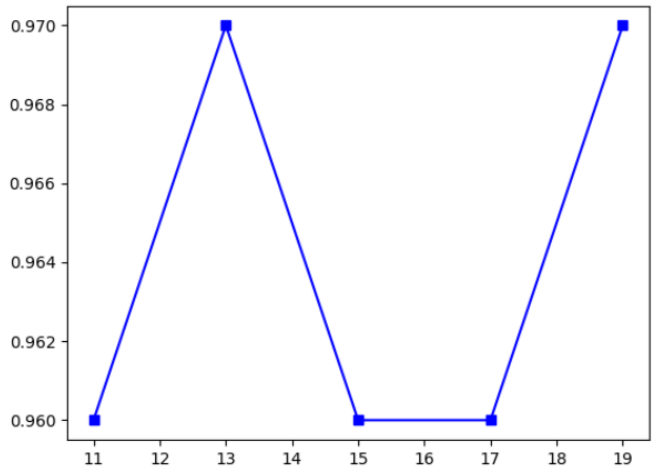
## 6. 학습 과정에 대한 설명

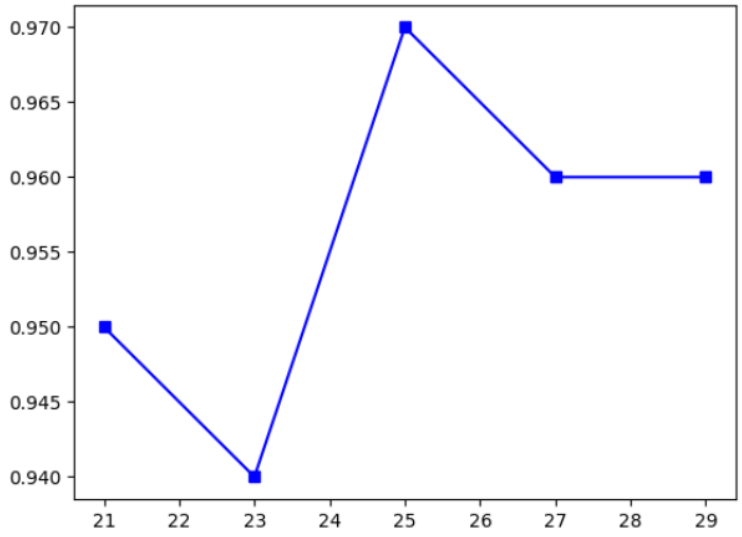
KNN은 단순히 트레인셋을 보유하는 것이 모델의 전부이다. 따로 학습을 진행하지 않는다. 사전에 학습된 결과로 부터 새로운 데이터포인트의 라벨을 도출하는 것과는 거리가먼 알고리즘이다. 따라서 KNN을 사용할때는 inference를 낮추는 방향과 정확도를 높이는 것을 목표(예를 들어 해당과제의 적절한 k값 선정)로 방향성을 가져야한다.

## 7. 결과 및 분석

k값을 1부터 2단위로 증가시키며 연산한 결과이다.
x축은 k값을 y축은 정확도를 의미한다.






최고 정확도는 0.97로 k = 13, 25일때 가장 높은 정확도를 기록했습니다.
각각의 테스트는 5개의 k값을 사용하여 각자 실행되었습니다 평균적으로 k값 한번당 300ch 초의 inference가 발생했습니다.