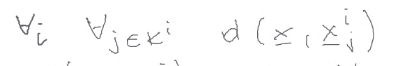
1. Umieć dokonać klasyfikacji (NN, k-NN, NM+Mahalonobis [odległość]) (1 - 2 wykład)
2. Klasyfikacja NN (Nearest Neighbour) / (Najbliższego Sąsiada)
   * jeżeli wektory są n-wymiarowe to konieczna jest normalizacja danych
   * dla każdej klasy ‘i’ i dla każdego punktu ‘j’ obliczamy odległość między nieznanym punktem a ‘j’ punktem klasy ‘i’ (odleglosc L2, L1 lub L)



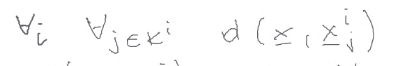
* + wybieramy najbliższy punkt z każdej klasy (bo jest nam tak daleko do klady jak do najbliższego jej punktu)



* + wybieramy tą klasę, dla której wynik jest najbliższy



1. Klasyfikacja k-NN
   * jeżeli wektory są n-wymiarowe to konieczna jest normalizacja danych
   * dla każdej klasy ‘i’ i dla każdego punktu ‘j’ obliczamy odległość między nieznanym punktem a ‘j’ punktem klasy ‘i’ (odleglosc L2, L1 lub L)

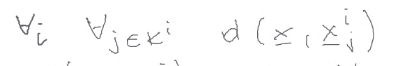


* + k - określa liczbę najlepszych elementów, najlepiej nieparzysta, dobieramy eksperymentalnie
  + obliczając jednocześnie odległości, uaktualniamy wektor zwycięzców
  + wybór klasy dominującej w wektorze zwycięzców

1. Klasyfikacja NM (Nearest Mean) / (Najbliższej Średniej) + Mahalanobis

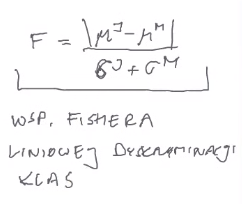
Trywialna obliczeniowo, niewrażliwa na złe przykłady (obserwacje odstające)

* + dla każdej klasy ‘i’ i dla każdego punktu ‘j’ obliczamy odległość między nieznanym punktem a ‘j’ punktem klasy ‘i’ (odleglosc L2, L1 lub L)

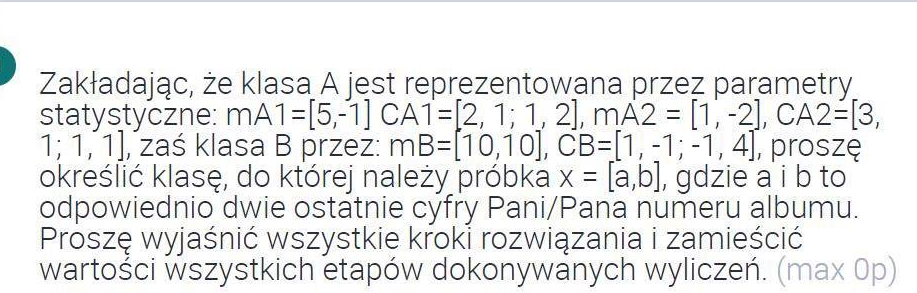


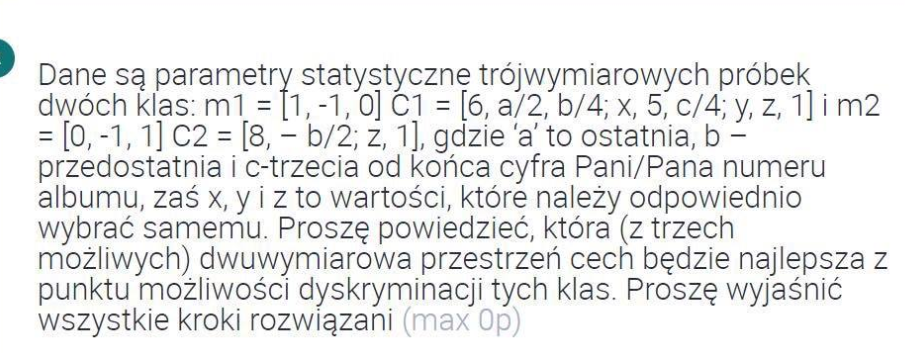
* + obliczamy średnią odległość z próbek dla danej klasy
  + obliczamy odchylenie standardowe dla klasy
  + obliczamy odległość Mahalanobisa

1. Policzyć jednowymiarowy współczynnik Fishera
   1. lepsza jest ta cecha, dla której współczynnik jest większy



ZADANIA





mA1 = [5, -1]

CA1 = [2,1 ; 1, 2]

mA2 = [1, -2]

CA2 = [3, 1; 1, 1]

mB = [10, 10]

CB = [1, -1; -1, 4]

x = [3, 8]

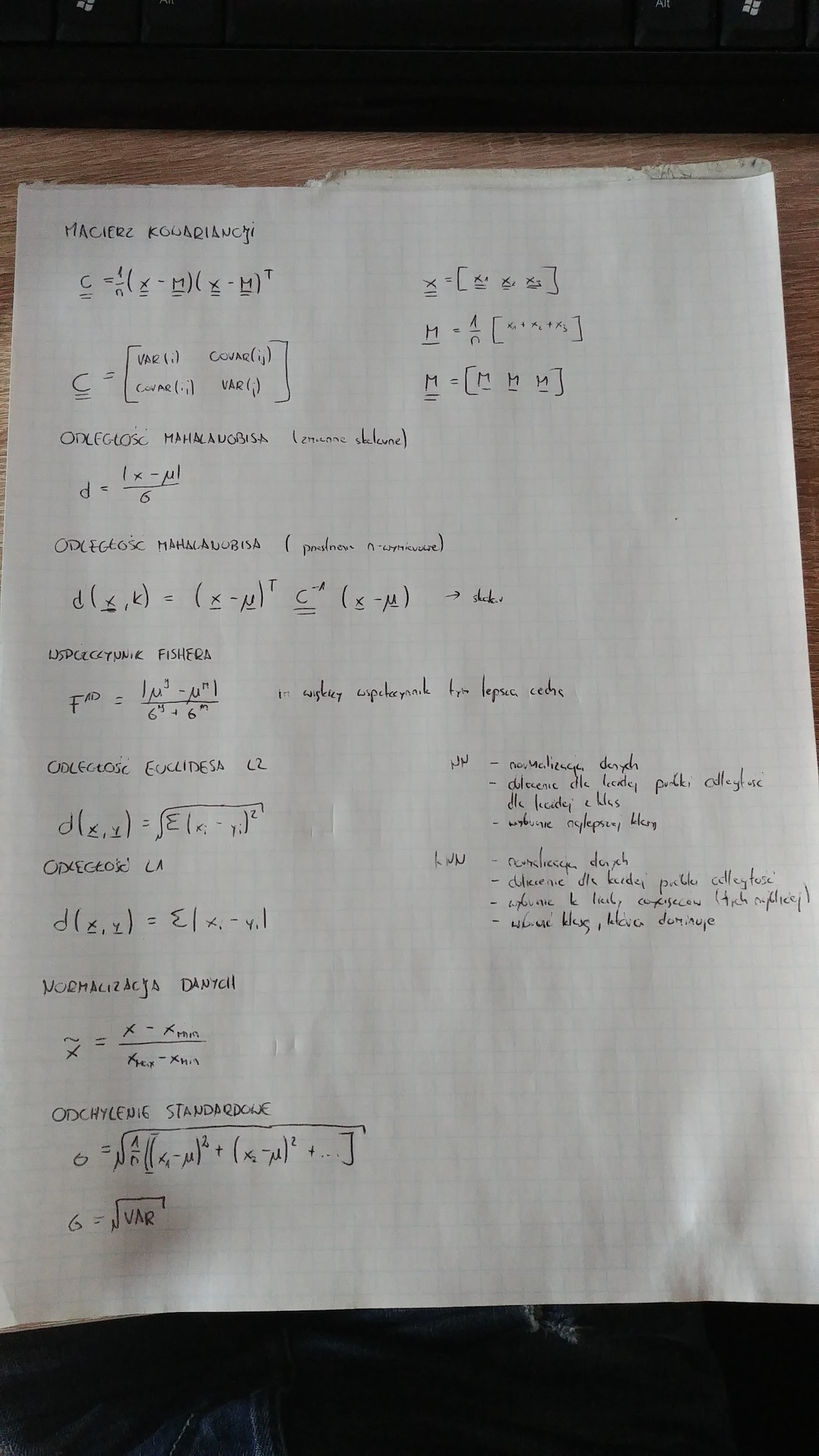
CA1-1 = 

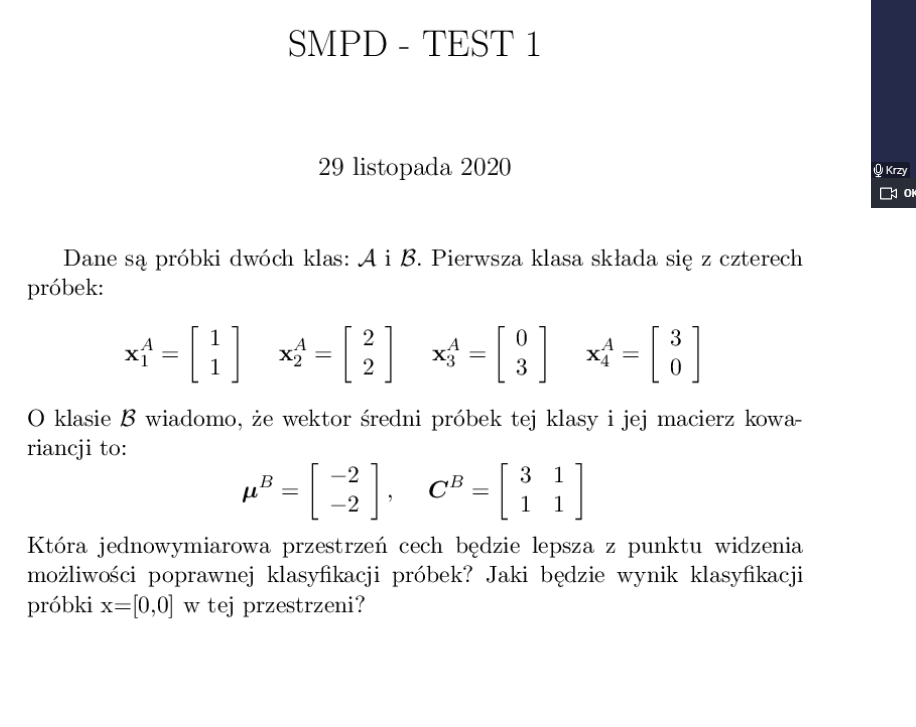
CB-1 = 

d(x, KA)=[-2, 9]T\*CA1-1\*[-2 , 9] = [-13/3, 20/3|T \* [-2, 9] = 206/3

d(x, KB)=[-7, 2]T\*CB-1\*[-7 , 2] = [-26/3, -5/3] \* [-7, 2] = 172/3

x należy do Kb.

**j = 1 ez**

****