Q1 1
2 Points

"A computer program is said to learn from experience E with respect to some task T ans some performance measure P, if its performance on T as measured by P, improves with experience E." Suppose your email program watches which e-mails you do or do not mark as spam, and based on that learns how to better filter spam. What is the task T in this setting?

- ○ Watching you label emails as spam or not spam
- ○ The number of emails correctly classified as spam/not spam
- ◉ Classifying emails as spam or not spam
- ○ None of the above - this is not machine learning problem

Q2 2
2 Points

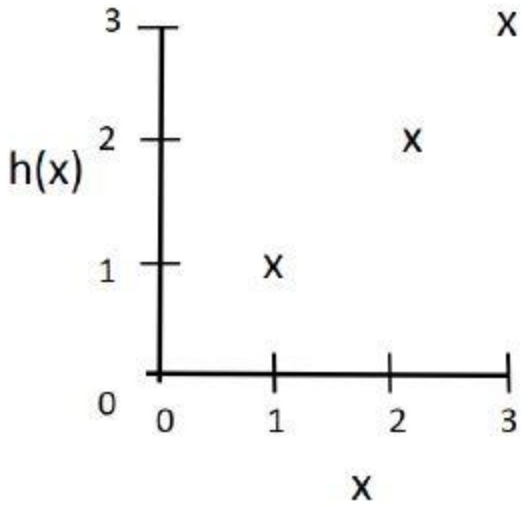
Suppose you are running a company and you want to develop learning algorithms to address two problems:
Problem 1: You have a large inventory of identical items. You want to predict how many items will sell over the next 3 months.
Problem 2: You'd like the software to examine individual customer accounts, and for each account decide if it has been hacked/compromised.

- ○ Treat both as classification problems.
- ○ Treat problem 1 as a classification problem, problem 2 as a regression problem.
- ◉ Treat problem 2 as a classification problem, problem 1 as a regression problem.
- ○ Treat both as regression problems.

Q3 3
5 Points

Suppose we have training set with m=3 examples, plotted below. Our hypothesis representation is $h(\theta) = \theta_1 x$ with parameter θ_1. The cost function is $J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$.

What is $J(0)$?



No files uploaded

Q4 4

1 Point

Which of the following are true statements? Select all that apply?

- [x] To make gradient descent converge, we must slowly decrease over time
- [ ] Gradient descent is guaranteed to find the global minimum for any function $J(\theta)$.
- [ ] Gradient descent can converge even if α is kept fixed. (But α cannot be too large, or else it may fail to converge.)
- [ ] For the specific choice of the cost function, J used id linear regression, there are no local optima (other than the global optimum).

Q5 5

15 Points

The values of y and attribute x are given in the table below:

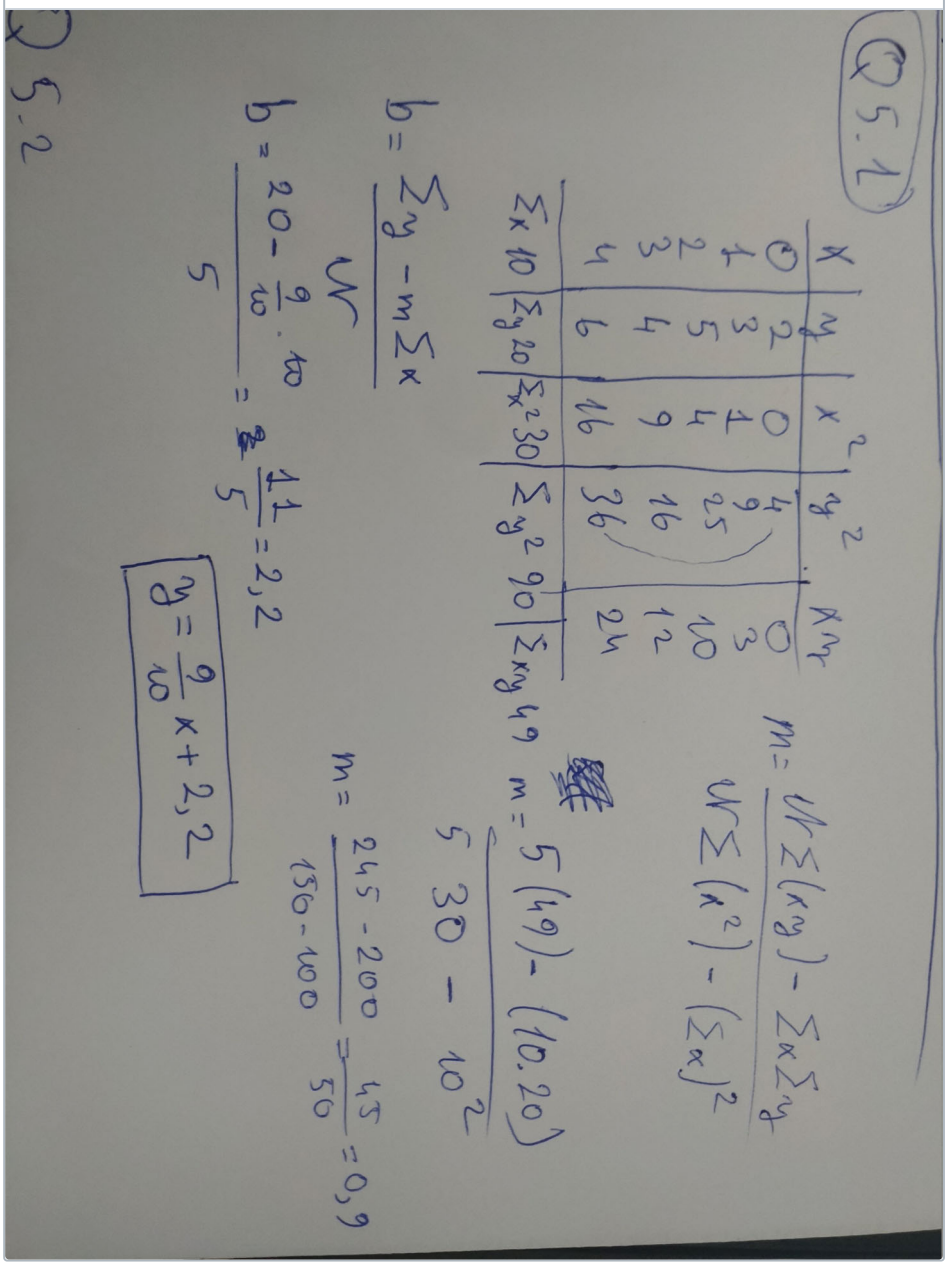
x	0	1	2	3	4
y	2	3	5	4	6

Q5.1 a

10 Points

Find the least square regression line y = mx +b

IMG_20210312_162758_N.jpg Download

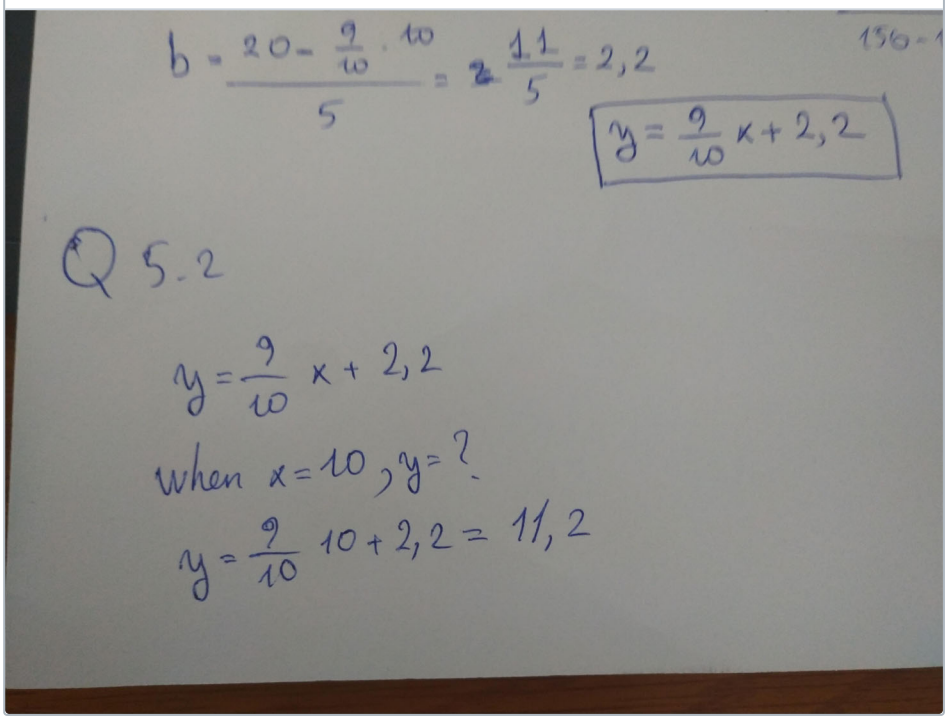


Q5.2 b

5 Points

Estimate the value of y when x= 10

IMG_20210312_162813.jpg Download



Q6 6

5 Points

Suppose we set $\theta_0 = -2, \theta_1 = 3$ in the linear hypothesis from Q1. What is $h(5) = ?$

243/16

Q7 7

70 Points

The problem of classification of Power Saving Lights by their economical feasibility as Preserver or Wasteful will be solved with k-NN. Suppose k = 3 and attributes : X_1 Lighting Duration and X_2 Power Consuming.

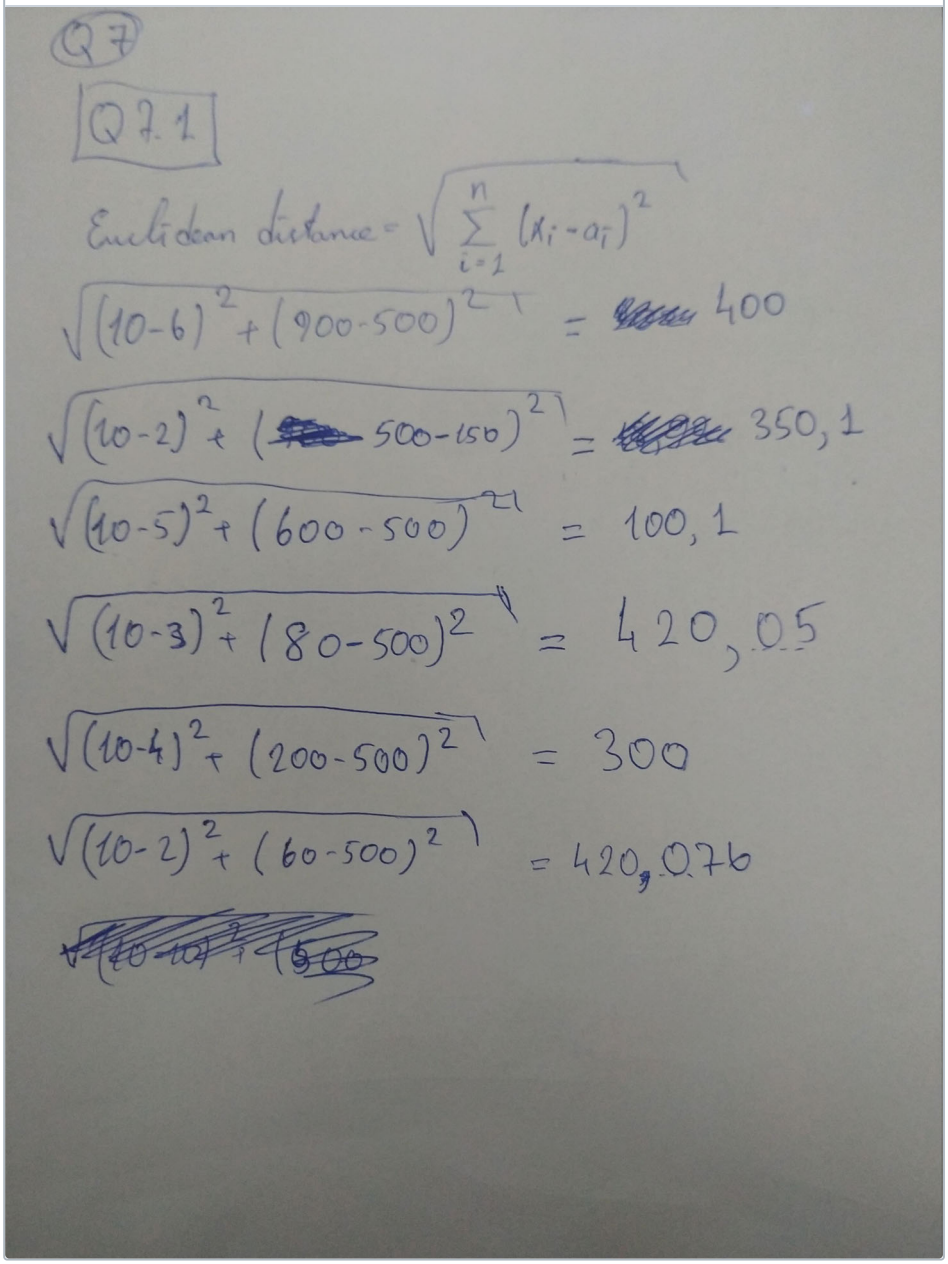
X_1 : Lighting Duration (Hours)	X_2 : Power Consuming (Watts)	Y
6	900	Wasteful
2	150	Wasteful
5	600	Wasteful
3	80	Preserver
4	200	Wasteful
2	60	Preserver
10	500	?

Q7.1 a

25 Points

Calculate the Euclidean distance between the test sample (X_1 : 10, X_2 : 500) and all training samples.

▾ IMG_20210312_165511.jpg Download

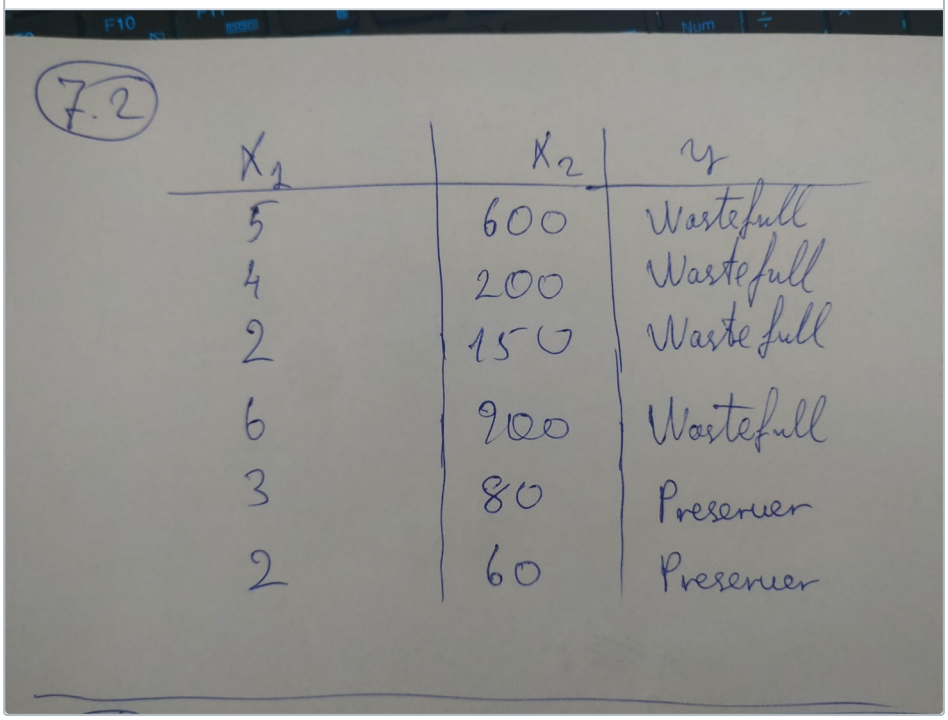


Q7.2 b

5 Points

Sort the training samples based on the calculated distance.

IMG_20210312_165533_N.jpg Download

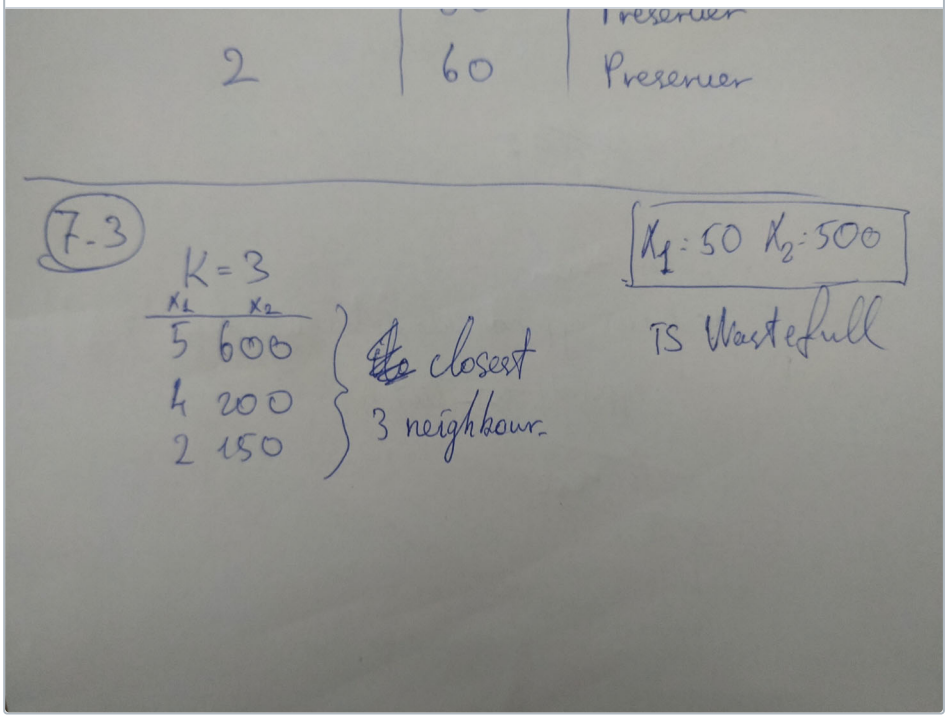


Q7.3 c

10 Points

Find the label of the test sample based on the closest k neighbors (k:3)

IMG_20210312_165538_N.jpg Download



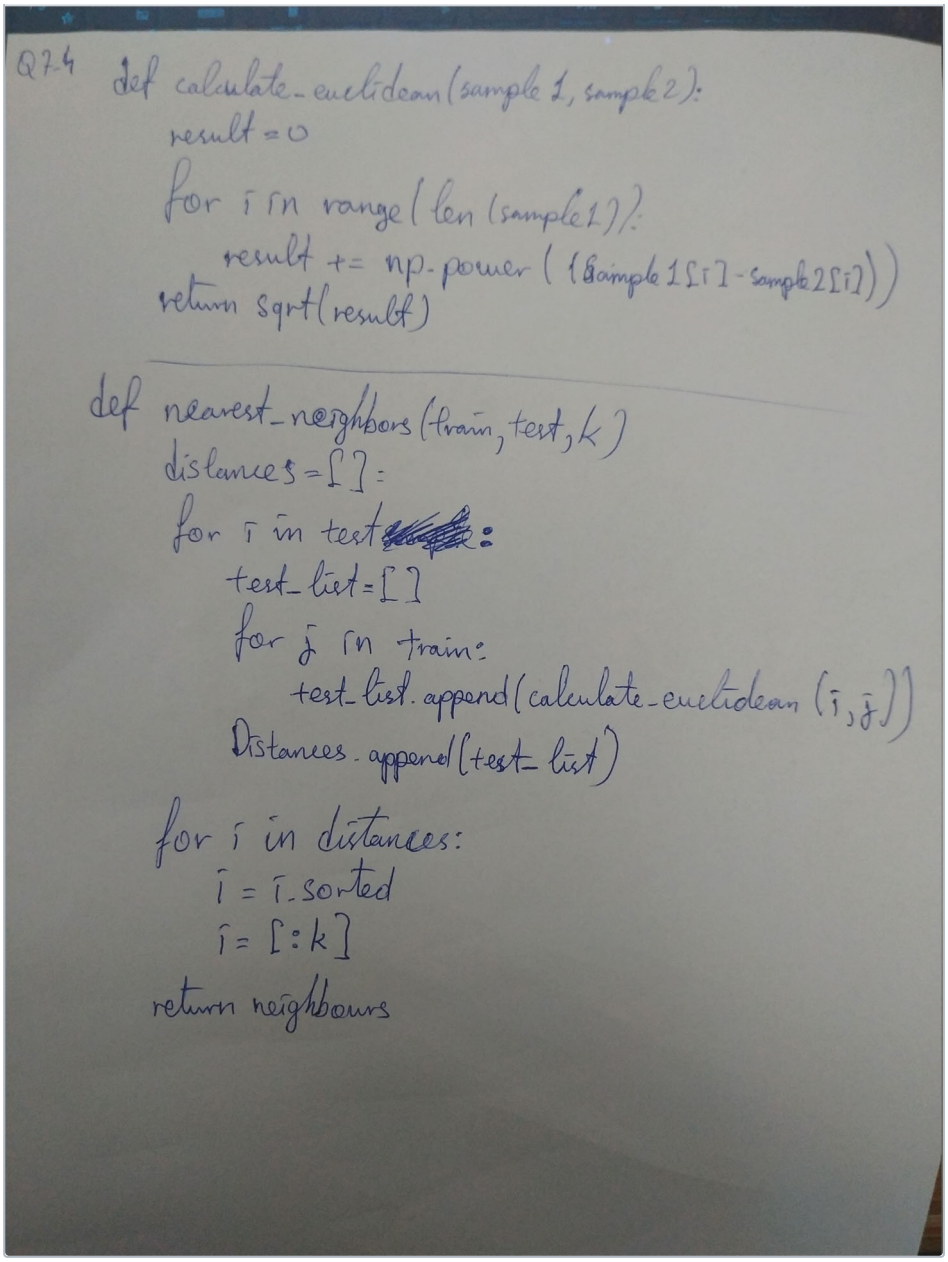
Q7.4 d

30 Points

Fill in the functions for the given tasks.

```
def calculate_euclidean(sample1, sample2):



    return distance

def nearest_neighbors(train_set, test_sample, k):
    \\calculate distances between test_sample and train_set



    \\find closest neighbors and return k neighbors



    return neighbors

def predict(train_set, test_sample, k):
    neighbors = nearest_neighbors(train_set, test_sample, k)



    return label_test_sample
```

▾ IMG_20210312_165552.jpg Download



```
Q7-4 def calculate_euclidean(sample1, sample2):
        result = 0
        for i in range(len(sample1)):
            result += np.power((sample1[i] - sample2[i]))
        return sqrt(result)

     def nearest_neighbors(train, test, k)
        distances = []:
        for i in test:
            test_list = []
            for j in train:
                test_list.append(calculate_euclidean(i, j))
            Distances.append(test_list)
        for i in distances:
            i = i_sorted
            i = [:k]
        return neighbours
```

Quiz-1

GRADED

STUDENT
MEHMET TAHA USTA

TOTAL POINTS
84 / 100 pts

QUESTION 1

1		**2** / 2 pts
QUESTION 2		
2		**2** / 2 pts
QUESTION 3		
3		**0** / 5 pts
QUESTION 4		
4		**0** / 1 pt
QUESTION 5		
5		**15** / 15 pts
5.1	a	**10** / 10 pts
5.2	b	**5** / 5 pts
QUESTION 6		
6		**0** / 5 pts
QUESTION 7		
7		**65** / 70 pts
7.1	a	**25** / 25 pts
7.2	b	**5** / 5 pts
7.3	c	**10** / 10 pts
7.4	d	**25** / 30 pts