***Exercise 5: Theory + SVM***

Or Livne – 203972922

Roy Rubin – 201312907

Text, letter

Description automatically generated

1. Q1 – kernels and mapping functions
   1. Finding the mapping function:
      * Reminder: a function K is called a kernel if there exists a mapping function so that the following holds:
      * Note: in ex.1.a., K is defined over which means that x and y are vectors of size 2.
      * Using the hint:

So, what we are actually looking for is:

* + - Definition of :
    - Proof:

I note that something still doesn’t add up because x and y are 2d vectors.

* 1. I am not sure … Monomial ? full variational? Or something ?
  2. How many multiplications are saved ?
     + Note that is an inner product of two vectors of dimension 10 (!)
     + Note that is a multiplication of vectors of dimension 2 (!)
     + The number of multiplications in the higher dimension is: 100 ?? (Should be exact)
     + The number of multiplications in the original dimension is: 8 ??

Graphical user interface, text, application

Description automatically generated

1. Q2 - Solving with LaGrange multipliers:
   * + Finding the maximum and minimum points for the function f under the constraint g: we will extract from the equation: which is equivalent to:

Meaning we can extract:

* + - Finally, we will solve a 3-way equation with the help of the original given constraint:

This will give us 2 values. We will extract them, and use them to find the min and max points:

* + - We received:
    - When analyzing the function f, we note that the possible point that will maximize f will occur when x is positive and y is negative, and the possible point that will minimize f will occur when x is negative, and y is positive.
    - So, the maximum point of the function f will occur when:
    - The maximum point of the function f will occur when:

Text, letter

Description automatically generated

* + - Let , Let , w =
    - Let
      * be the set of all origin-centered upright equilateral triangles.
    - **We needed to:**
      * Describe a polynomial sample complexity algorithm 𝐿 that learns 𝐶 using 𝐻.
      * State the time complexity and the sample complexity of our suggested algorithm.
      * We needed to prove all our steps.

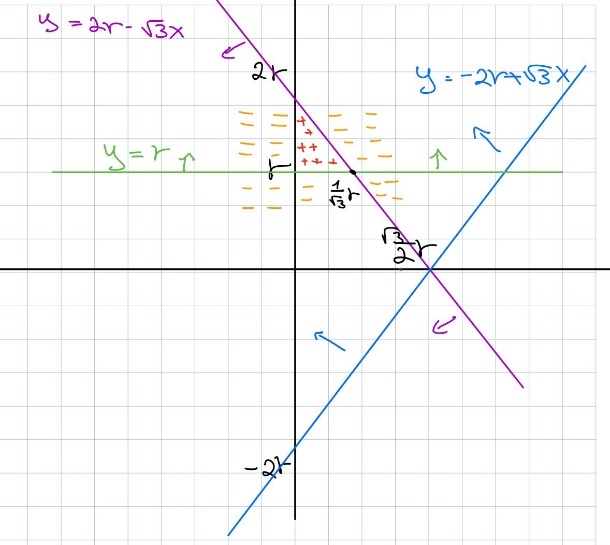
**Answer:**

* + - The algorithm will produce a hypothesis which is the smallest relevant origin-centered upright equilateral triangles that contains all the positive points.
    - This can be done in as follows:
      * Let be a set of points in the 2D space, labeled positive and negative.

Our algorithm seeks to return a hypothesis ℎ ∈ 𝐻.

* + - * Base condition there are 3 equations:

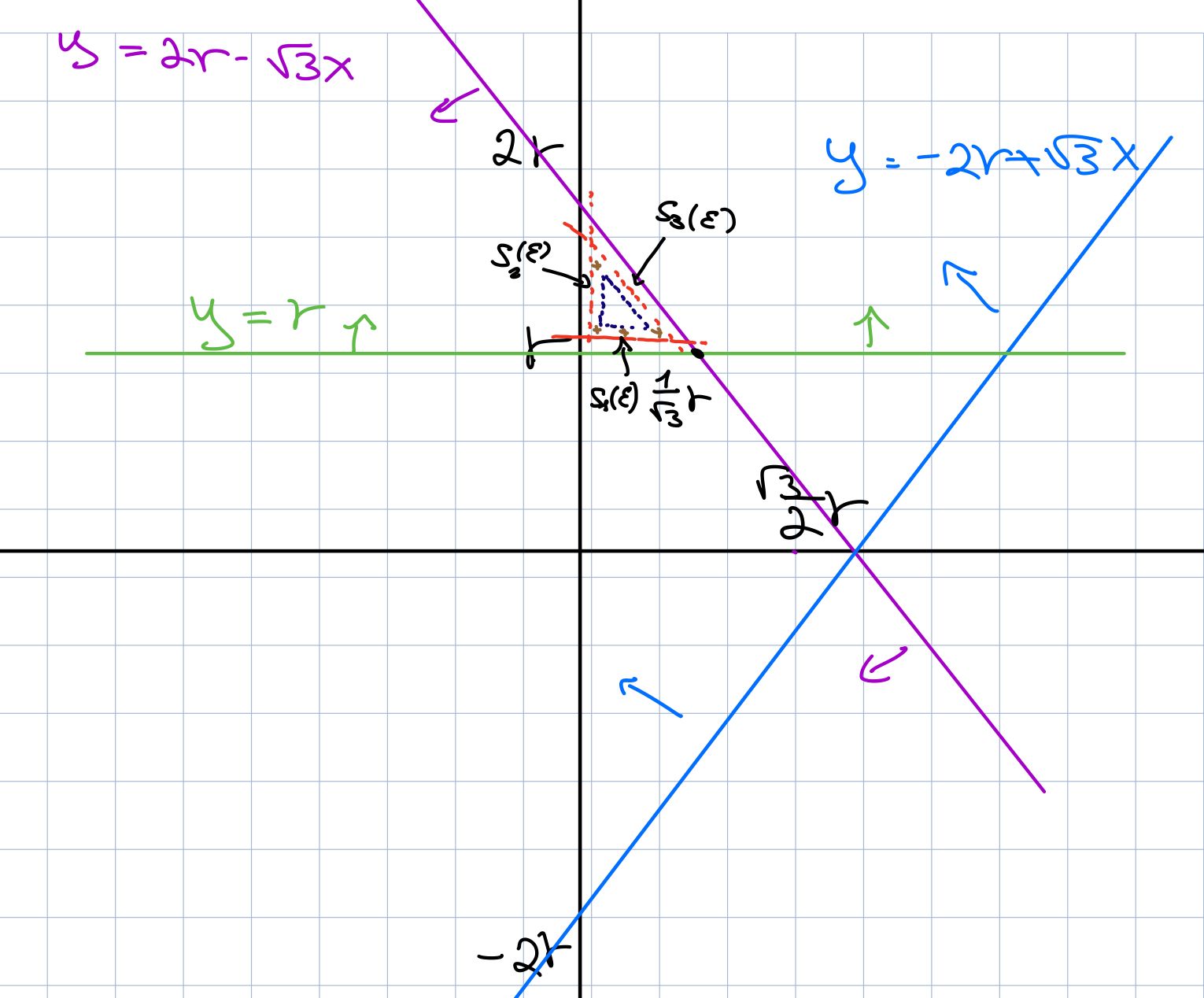
1. Figure of results base the conditions:



* + - Base the figure above, let be all positive labeled data points, which means
* + - The vertices of the hypothesis triangle will be
    - Consider be training data generated from c without errors and by drawing m independent points according to a probability distribution in We will denote the probability distribution thus induced on () by
    - Given , we will now compute a number m () so that **Eq1**:

🡪 e() =

* + - Note that is the hypothesis h, or the triangle, produced by L when considering data as above.
    - e () is a random variable that depends on the stochastic behavior of
    - it is exactly this behavior that we will want to characterize.



* + - Consider the strips parallel to the edges of the triangle c as in figure above, these are defined to satisfy:
    - Now not that:
    - Where D: =
    - This because if visits all three strips (note that negative points cannot visit the strips as there are no errors) then, according to our construction, the difference between c and .
    - In term of probability, we therefore get:
    - Now we will select m () = to get **Eq 1** to hold.
* =
  1. The maximal product between point in X to the v vector from the origin
  2. The maximal product between point in X to the w vector from the origin
  3. The maximal product between point in X to the u vector from the origin
* =
  1. The maximal product between point in X to the v vector from the origin
  2. The maximal product between point in X to the w vector from the origin

Simply, this is an origin-centered equilateral triangles where the sizes r,r are the maximal distances we’ve seen in the training set.

* + - * It spans from , this box is contained in the ground-truth box (concept c).

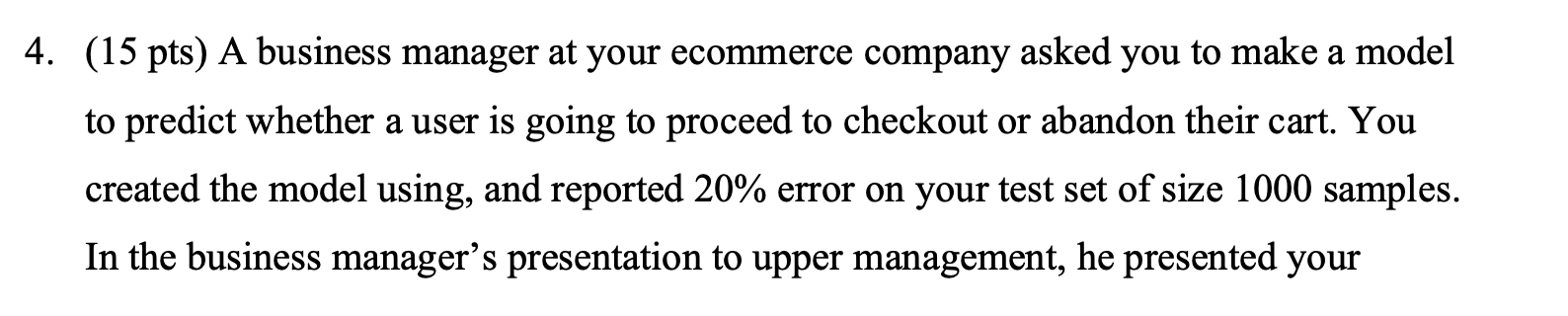
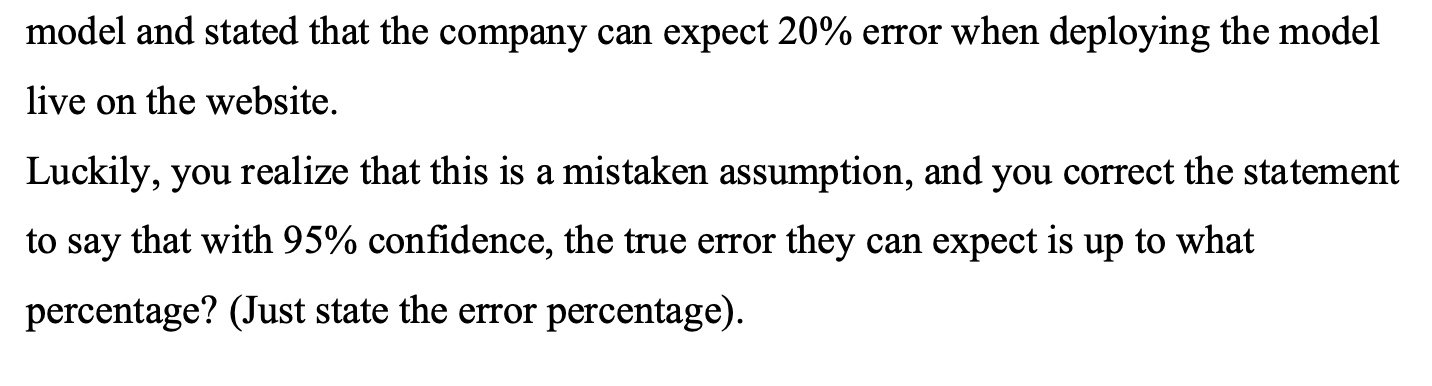
h:

* Time complexity is: O(m) for each vector 🡪 Total O(2m) = **O(m)**
  + - Now we consider the area between our h to c (remember that ).

There are \_\_\_ such areas:

* + - * for each of the coordinates
    - Consider training data, .
    - Assume that D visits each one of the 2 sets 𝐵, defined above.
    - What can we say about Err(ℎ, 𝑐)?
    - So, the probability of a point in to be in **either** of those areas B\_i is
    - For a given 𝜀 and 𝛿, the number of samples needed is:

* + - Meaning, when we want a confidence of to get an error of , we will need **at least** training instances.



* + - We know that:
      * Base current model there is 20% error base m = 1000
      * We wish for 95% confidence
    - We assume that there is 10 attributes
    - Base the following equation:

Text

Description automatically generated

Text

Description automatically generated From inside the notebook:

1. Q5 – SVM with jupyter notebook
   * + My chosen C values: [0.0005, 0.001, 0.002, 0.005, 0.01, 0.1, 10]
     + Final graph in comparison to needed graph:

Chart, line chart

Description automatically generated