Machine Learning

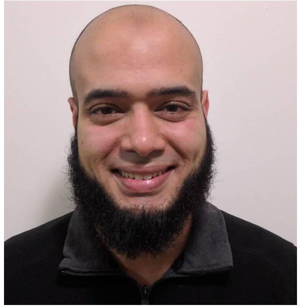
Regression for Binary Classification

Mostafa S. Ibrahim

Teaching, Training and Coaching for more than a decade!

Artificial Intelligence & Computer Vision Researcher

PhD from Simon Fraser University - Canada
Bachelor / MSc from Cairo University - Egypt
Ex-(Software Engineer / ICPC World Finalist)



Recall: Machine Learning Algorithms

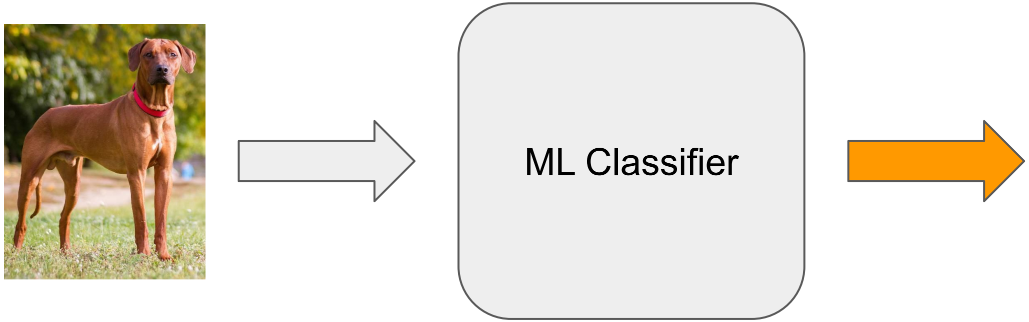
- In each learning paradigm, a diverse range of different algorithms are employed, each aligned with a specific learning approach:
- Supervised learning
  - Regression (Linear/Polynomial, **Neural Network,** Decision Trees, Random Forest, **XGBoost**)
  - Classification (Logistic Regression, **Neural Network,** KNN, Naive Bayes, SVM, Decision Trees, Random Forest, **XGBoost**)
  - Recommender (Collaborative filtering, Content-based filtering: can use **Neural Network**)
- Unsupervised learning
  - Clustering (K-means, GMM, **Neural Network**)
  - Dimensionality Reduction (تخفيض الأبعاد)(**Neural Network**, PCA, t-SNE)
  - Generative modeling (Deep **Neural Network**: GAN, VAE)
- Reinforcement learning
  - State of the Arts (SOTA) uses deep **Neural Network**

Classification Task

- Classification is a **supervised** learning task where the output is a class label
- Class/**Discrete** labels examples:
    - 0 or 1 (for example, spam or not spam)
        - We call it **binary** classification
    - Labels from 0 to 4, representing categories like [cat,dog,person,car,bicycle]
        - We call it **multiclass** classification
- The number of classes is typically limited, often fewer than 50.
    - However, in deep learning applications, it can extend to thousands.

Classification Task

- Imagine we've trained a **classifier** to **classify** an image into: cat, dog, person, car, or bicycle



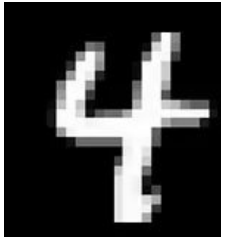
ML Classifier

- This is a **dog** (with 80% certainty)
- This is a cat (with 10% certainty)
- This is a person (with 5% certainty)
- This is a car (with 2% certainty)
- This is a bicycle (with 3% certainty)

- The classifier assigns a probability for each output class/category
  - Observe: the probabilities above sum to 1 (100%)
- We can select the class with the **highest probability**, such as the dog in this instance

Question!

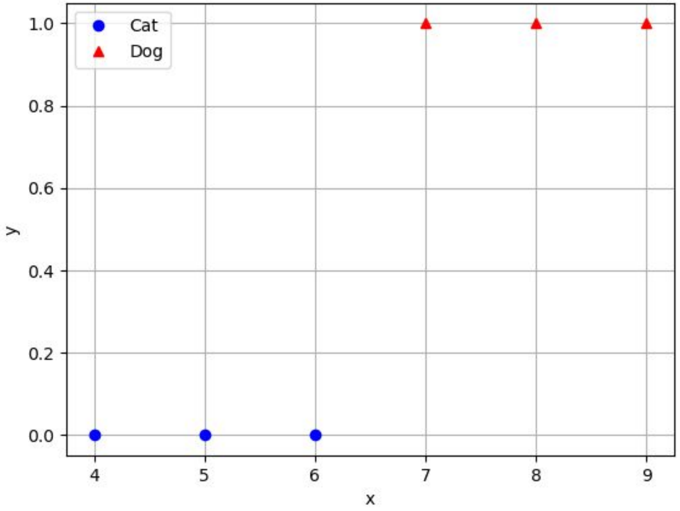
- Identify the classification tasks among the following:
- 1: Email: Is it spam or not?
- 2: Tumor (ورم): Benign (حميد) vs Malignant (خبيث)
- 3: Image is: Cat, Dog, Cow or Rat?
- 4: Transaction: Is it fraudulent? Yes or no
- 5: Judging a document: Sports, politics or education?
- 6: Eligibility for Loan? Yes or no?
- 7: Binary Image: Given a small image of a single digit in the range [0-9], identify it.



Dataset labeling

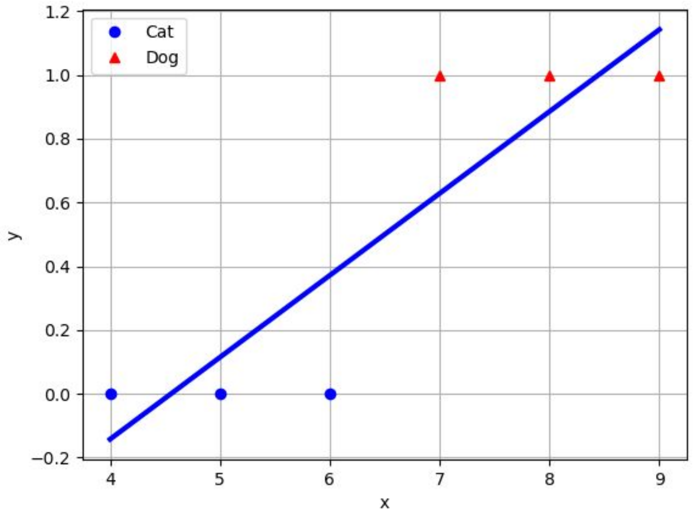
- We typically start labeling the classes in a random order
- The indexing follows: 0, 1, 2, 3, 4, ..., N-1
- For a binary problem, we have: 0 or 1
  - Based on the task type, we can **logically** select which class should be assigned to 1:
    - Is it spam?: Spam = 1, Not spam = 0
    - Tumor: Benign (Negative) = 0, Malignant = 1

Question!



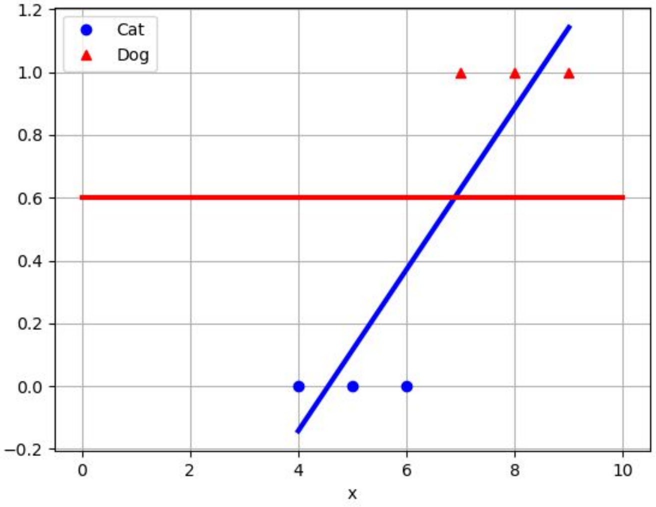
- Assume we trained a linear regression on this data
- 1) Where will the line be?

Answer!



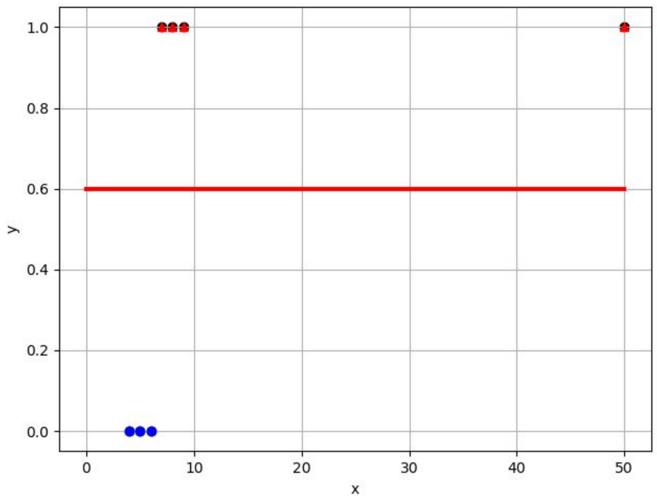
- The line aims to pass near the center of the data points.
  - (Note: Each point's distance is the projection onto the line.)
- 2) Given an input x, what is the possible output **range** for the prediction?
  - *It is open-ended, encompassing values less than 0 and greater than 1.*
  - 3) How can we **convert** the predicted value to one of the 2 classes?

Question!



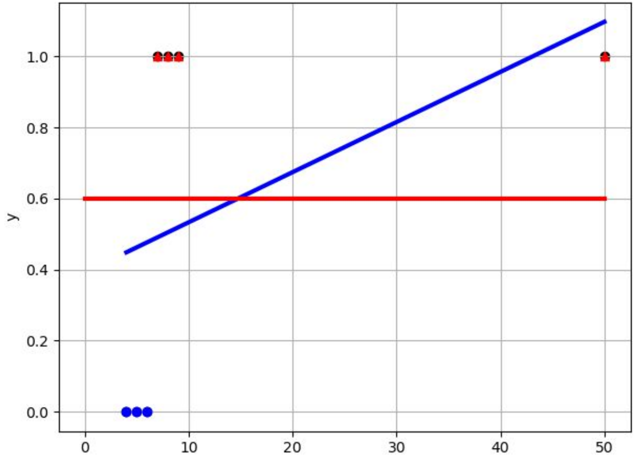
- We set a threshold, such as 0.6
- If the value bigger than 0.6, it's considered closer to the "1" region
  - Hence it's classified as '1'
- Otherwise, it's classified as 0
- In the visual representation, all training points fall into their respective classes with correct classification

Question!



- Assume we added a single point at x = 50, e.g. an **outlier**
- What potential effect could this have on the regression line?
- What potential effect could this have on the regression line?

Question!



- The line will be shifted severely to not be so far from the added point
- The first 3 red points are now misclassified (before threshold)
- Summarize 2 points that represents why we shouldn’t use regression for classification?

Regression for classification

- 2 key problems:
  - Linear regression just tries to fit a line
    - It will find a line that is good for the data
      - Sensitivity to added points, especially outliers
    - By definition, this is so **different** from the classification goal!
  - Its output is open range [-OO, OO]
    - We can get scores so far from [0, 1]
- We need a solution that solves this 2 problems!
  - It has a **decision boundary** dedicated for classification not regression
  - Its output is **bounded** and relevant for binary labels [0, 1]

sklearn.datasets import load_**breast_cancer**

- This is a toy dataset for binary classification: benign vs malignant
- I tried the following 3 models (code attached)
  - **Classifier**: 97% test accuracy
  - **Linear regression**: 94% test accuracy
  - **Neural network for regression**: 92% test accuracy
- But again, this is a toy dataset! Don’t count on that for real problems
  - Regression might work on some simple binary classification problems but with suboptimal performance.

“Acquire knowledge and impart it to the people.”

“Seek knowledge from the Cradle to the Grave.”