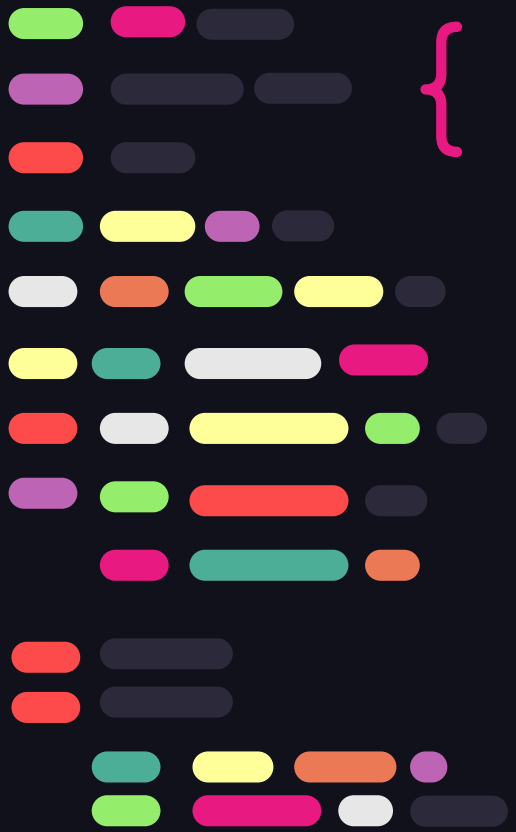


HackTricks 2024

Spaceship in the multiverse

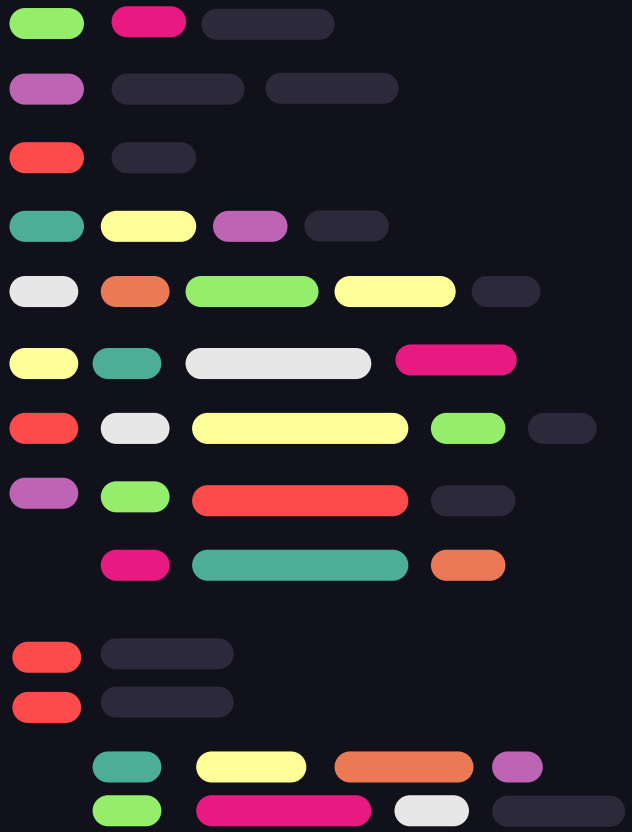
```
class SpaceshipInTheMultiverse:
    def __init__(self) :
        self.members = [“Mahmoud Elhusseni”,
                        “Karim Nady”,
                        “Mazen Elnahla”,
                        “Mahmoud Hosam”,
                        “Osama Hussein”]
```

Strategy

- Analyze the scoring function for both fox and eagle.
- Divide the riddles between 4 team members, while the remaining member write the API.
- Build preprocessing pipeline.
- Build different models to predict real and fake footprints.

Team Contribution

- Each team member carefully read the documentations.
- Analyze the scoring function for both fox and eagle.
- Listen to slack Channel.
- Attend mentorship sessions, record important information and ask our questions.
- Build API to submit.
- Try Different parameters configurations to enhance fox score.



{ ..

Fox

} ..

Fox - Scoring Function

$$Score = Done(\alpha 1 \frac{\sum(\frac{msg_{Real}+msg_{Fake}}{msg_{Real}+msg_{Fake}+msg_{Empty}})}{n_{chunk}} + \alpha 2(1-(\frac{\sum msg_{Fake}}{6}-1)^2) + \alpha 3(1-\frac{timeTaken}{timeOut}) + \alpha 4 \frac{budget_{remaining}}{20})$$

Variables to configure:

- $\sum msg_{Real}$
- $\sum msg_{Fake}$
- $\sum msg_{Empty}$
- $timeTaken$
- $budeget_{remaining}$

Fox – Scoring Function

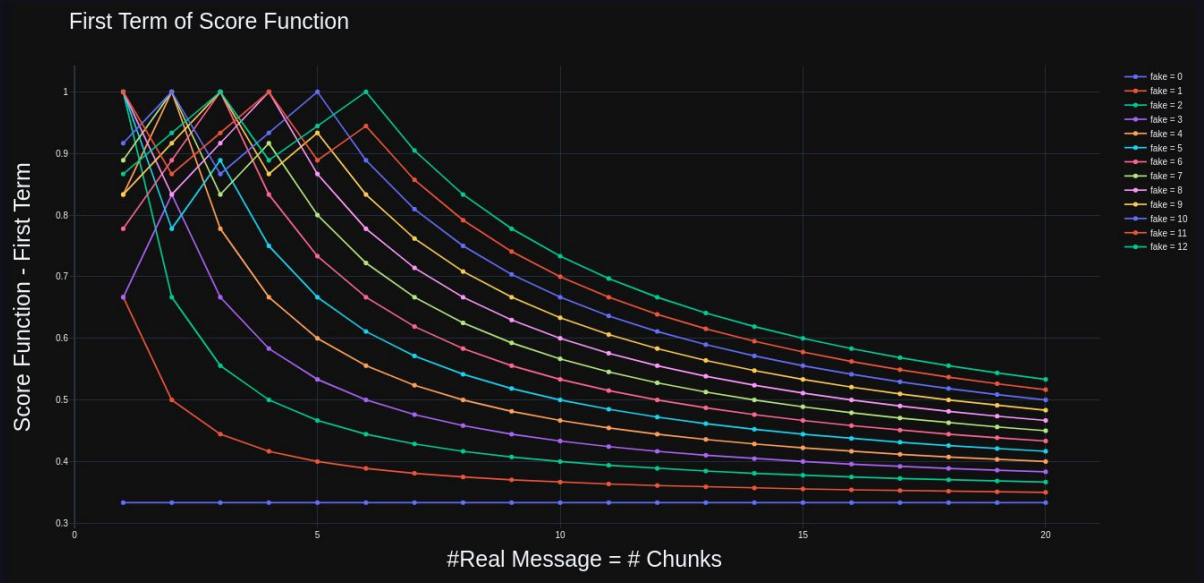
First Term:

$$t1 = \alpha 1 \frac{\sum\left(\frac{msg_{Real}+msg_{Fake}}{msg_{Real}+msg_{Fake}+msg_{Empty}}\right)}{n_{chunk}}$$

$$msg_{Real} + msg_{Fake} + msg_{Empty} = 3$$

$$t1 = \alpha 1 \frac{\sum(msg_{Real} + msg_{Fake})}{3 \times n_{chunk}^2}$$

$$\alpha 1 = 0.4$$

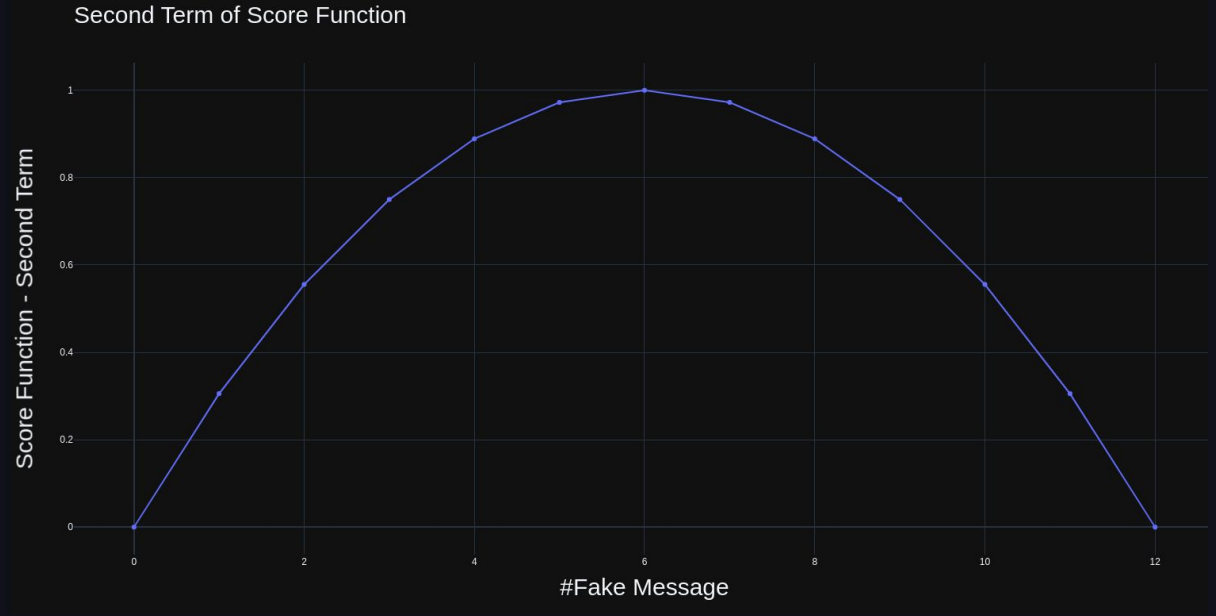


Fox – Scoring Function

Second Term:

$$t2 = \alpha 2(1 - (\frac{\sum msg_{Fake}}{6} - 1)^2)$$

$$\alpha 2 = 0.3$$

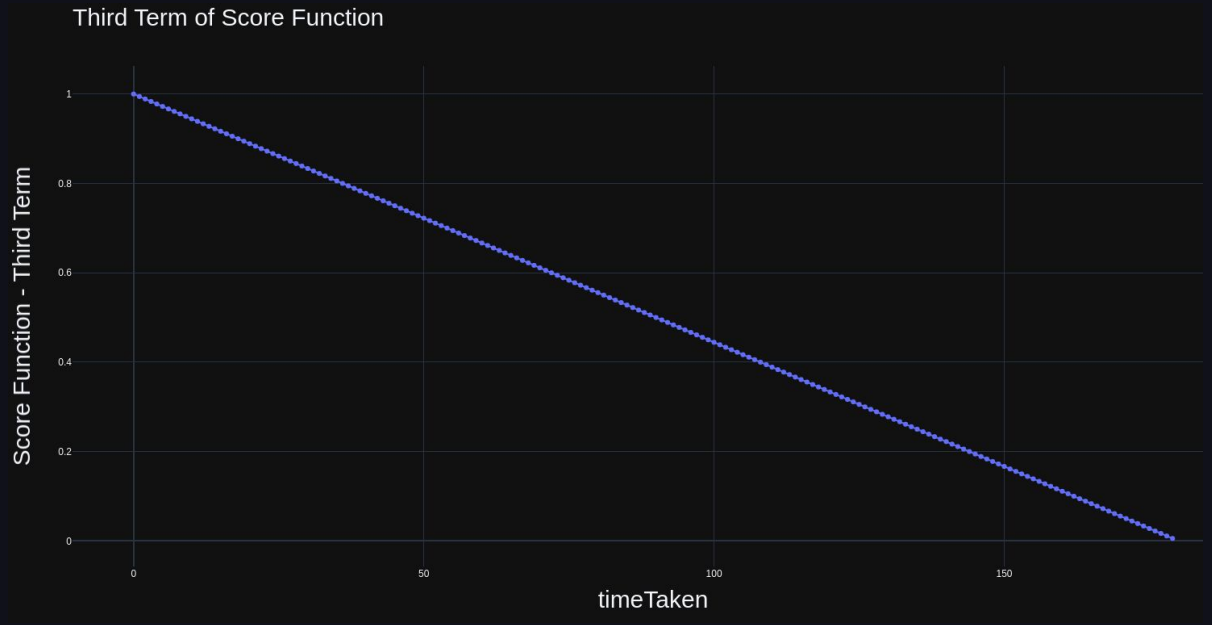


Fox – Scoring Function

Third Term:

$$t3 = \alpha3(1 - \frac{timeTaken}{timeOut})$$

$$\alpha3 = 0.2$$

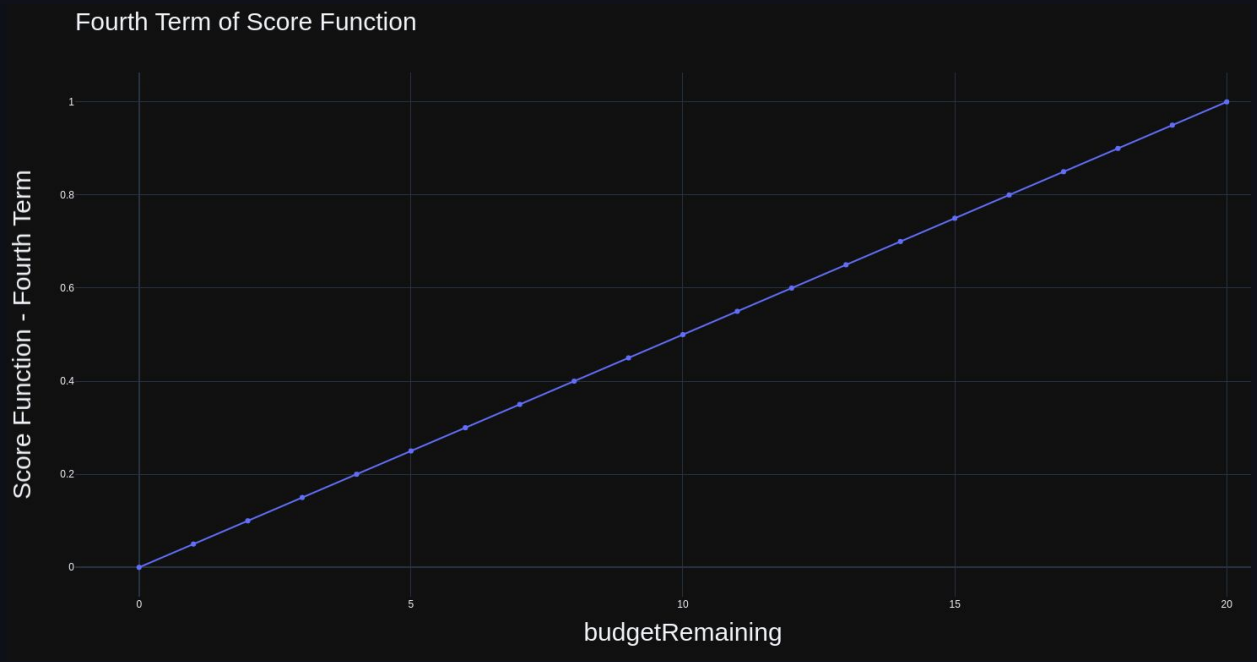


Fox – Scoring Function

Fourth Term:

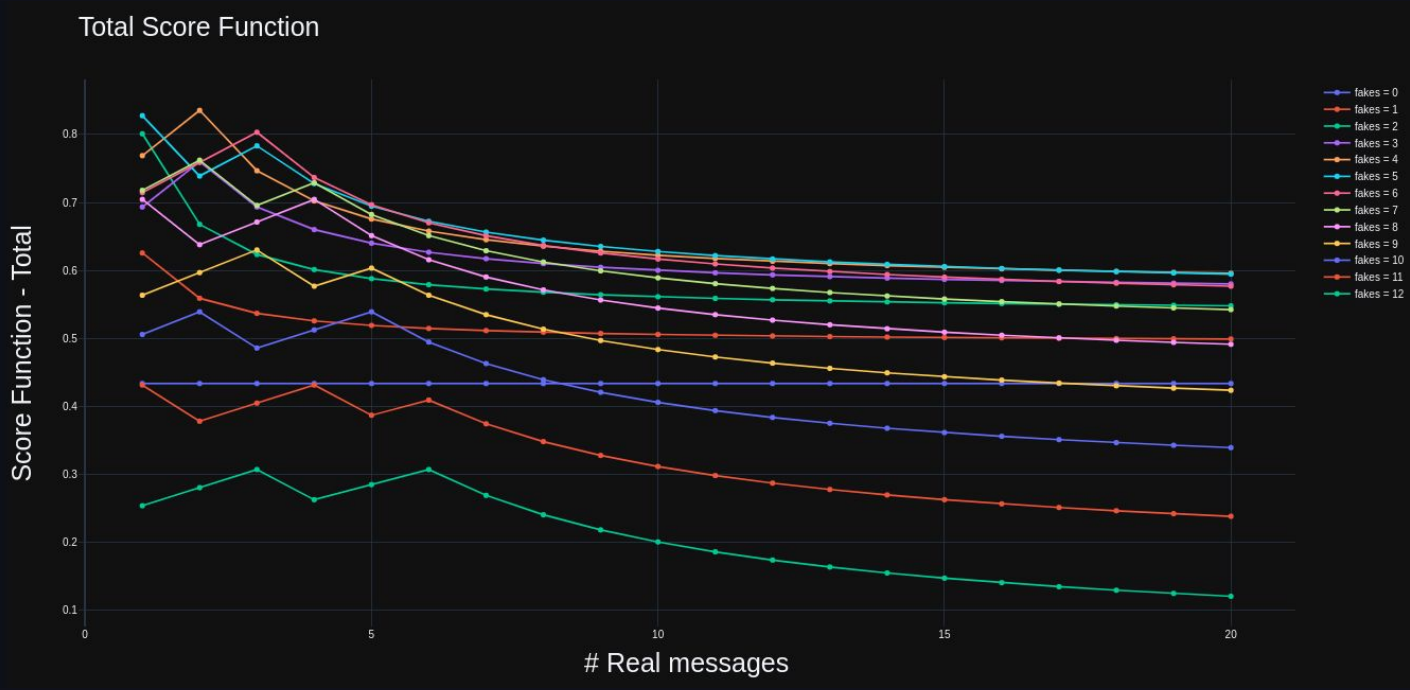
$$t4 = \alpha 4(\frac{budget_{remaining}}{20})$$

$$\alpha 4 = 0.1$$



Fox – Scoring Function

Overall

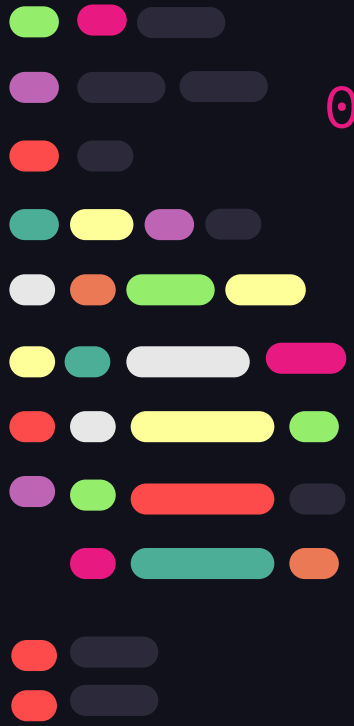


Fox - Approach

After Analyzing the Scoring Function we concluded some insights:

- Submit empty messages as few as possible.
- Number of fake messages submitted should be around 6 messages.
- Solve all riddles to maximize the budget.
- Solve only the less time consuming riddles.

Riddles



01 Cyber Security

- (Medium)
- (Hard)

02 Computer Vision

- (Easy)
- (Medium)
- (Hard)

03 Machine Learning

- (Easy)
- (Medium)

04 Problem Solving

- (Easy)
- (Medium)
- (Hard)

01 Cyber Security - (Medium)

SteganoGAN

Problem

- Finding the decoded message from an encoded image using the pretrained model.
- The given image given is a list of numbers represents the pixels of the image.

Our Solution

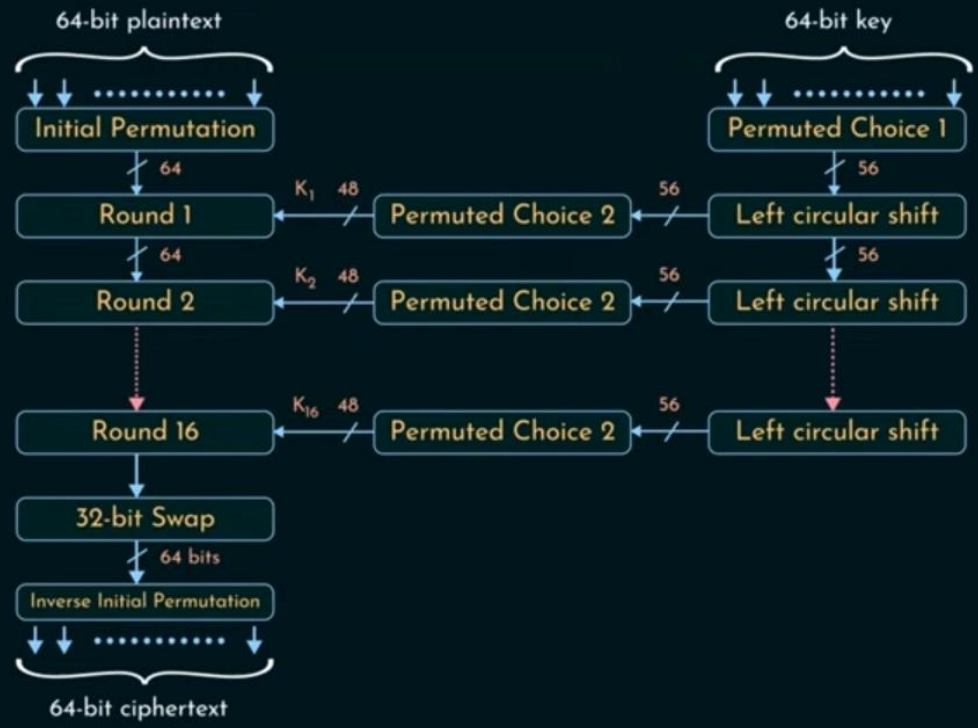
- Reading the image in RGB so the model can read it correctly.
- Convert pixel values to floating values.
- Convert the image from an array to tensors.
- Apply the tensors to the decode function

Enhanced Solution

- We found out that the given image list sometimes is doesn't need to be converted to floating numbers so we pass it to the decode function.
- we discovered that the model accepts pixel values from [1,0].
- So in real world case we need to read the image and divided the pixels by 255 and convert it to tensors

01 Cyber Security - (Hard)

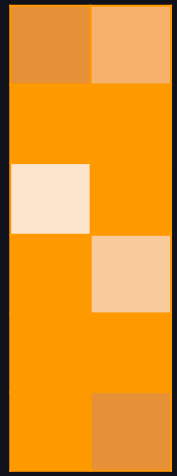
Data encryption standard DES Algorithm



02 Computer Vision - (Easy)

Fix the first shred in position.

Compare right edges for the current shred with the next potential shred left edge.



Determine the appropriate approach to fit edges together :

Approach 1 – Sum of Squared Differences (SSD)

Approach 2 – Counting Similar Pixels

02 Computer Vision - (Easy)

SSD

CSP

02 Computer Vision - (Medium)

Detect

Find the patch coordinates

SIFT extracts features from images

FLANN finds matches between these features

Mask

Create binary mask for the patch

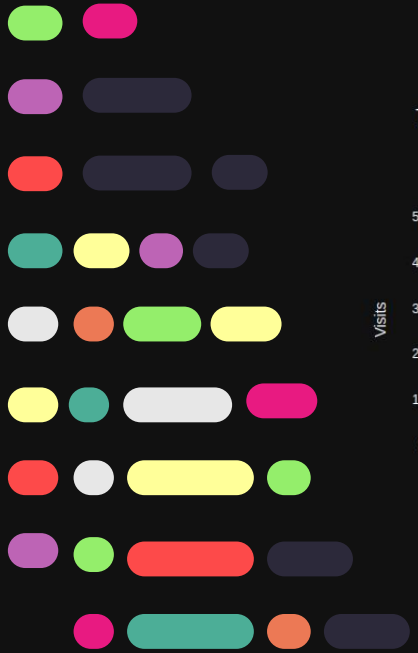
Using Perspective transformation to handle the rotation and scaling

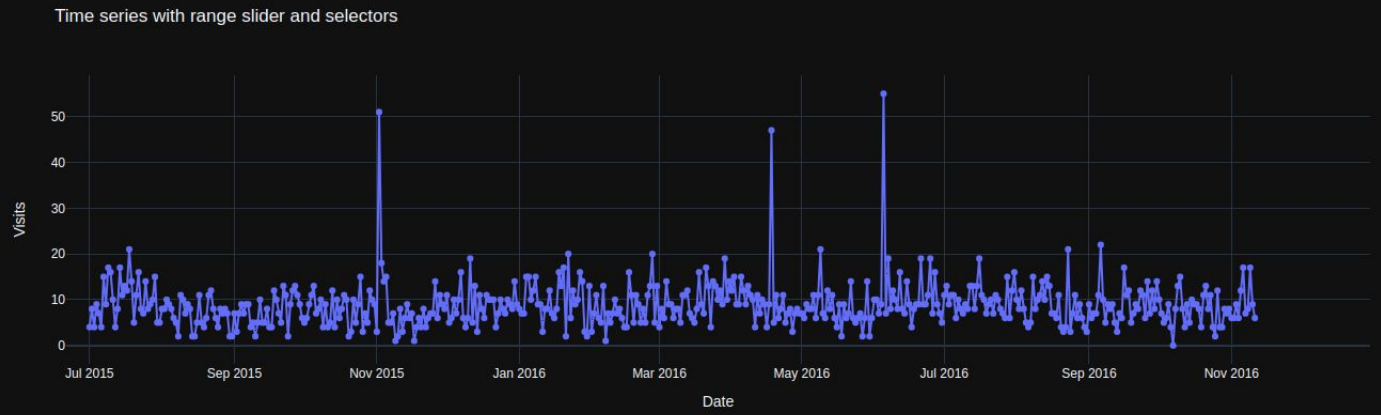
Paint

Use Telea Paint Algorithm

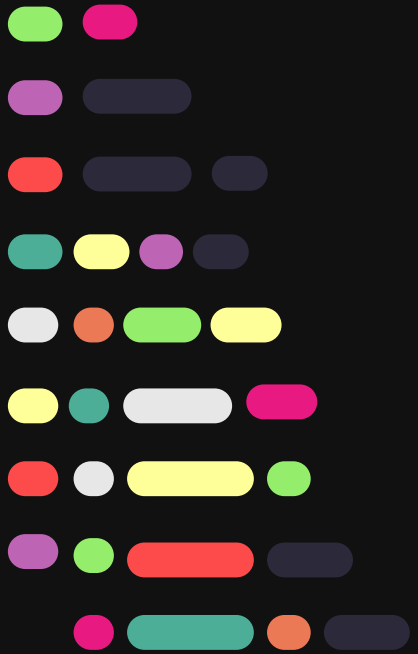
Use surrounding pixels to reconstruct the missing pixels

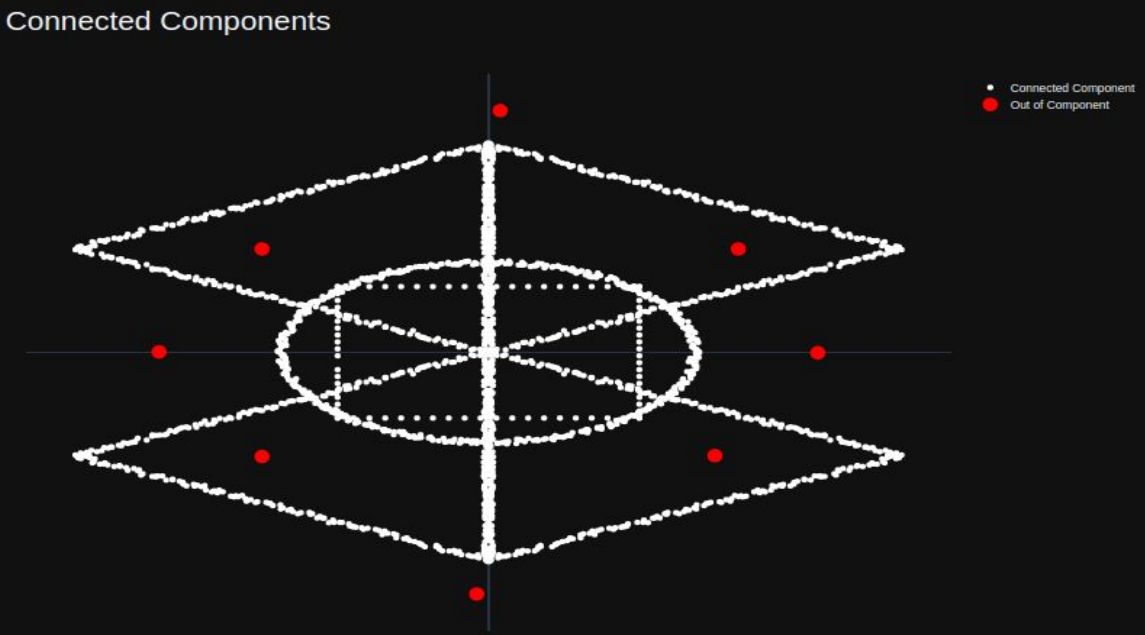
03 Machine Learning - (Easy)





03 Machine Learning - (Medium)





04 Problem Solving - (Easy)

Count Frequency of each word.

Create Frequency Pools contain all words with the same frequency.

Sort Pools Descending.

Sort Frequency each list Pools lexicographically.

Extract the Top X Words.

04 Problem Solving - (Medium)

Extract the number and word.

Push the word and number that we have if we find an open bracket inside Stack for each.

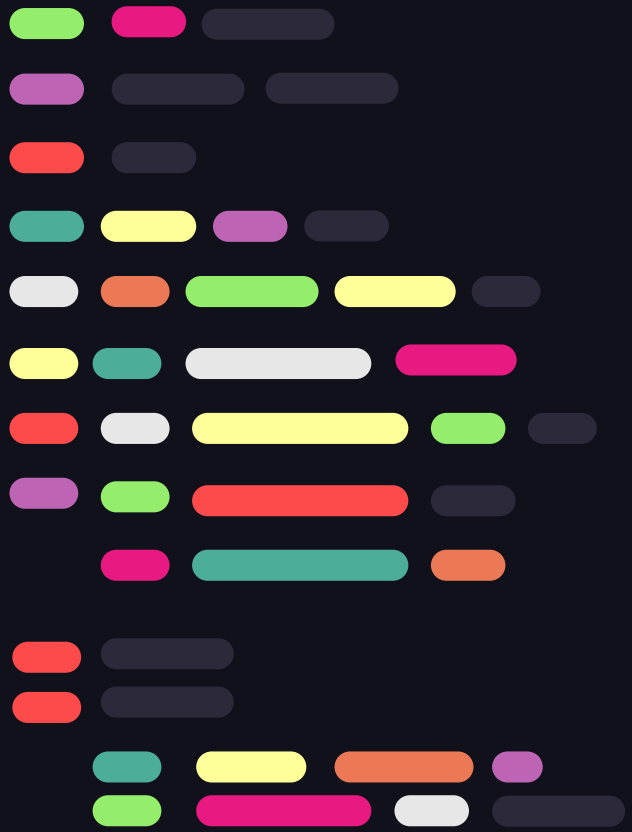
Pop the top of the Numbers Stack and multiply with word and add it to the top of Word Stack , if we have close bracket.

04 Problem Solving - (Hard)

Build Pre Calculated Table Answer

Calculate for each index in the table the number of ways to reach that index is the sum of the top and left index from it

1	1
1	2



{ ..

Eagle

} ..

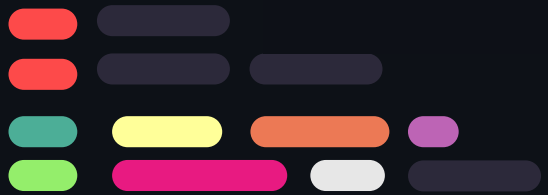
Eagle - Scoring Function

$$Score = (\alpha1 \times JaccardDistance + \alpha2 \times (1 - \frac{timeTaken}{timeTaken}))$$

$$Bonus = Score * 0.2 * \frac{Fake_{dodged}}{Fake_{chunks}}$$

$$Penalty = Score * 0.2 * \frac{Real_{missed}}{Real_{chunks}}$$

$$\alpha1 = 0.7, \alpha2 = 0.3$$



Eagle - Approach

First Approach - AI based approach

- Preprocessing spectrogram by
  - Replacing all infinity values with predefined value = 1e5.
  - applying logarithmic transformation to spectrogram.
- Simple CNN Network as encoder to extract features from melspectrograms.
- Two heads based decoder
  - First head to predict whether spectrogram represents fake or real message.
  - Second head to predict when “Dell” word has finished.

Eagle - Approach

First Approach - AI based approach

```
38it [00:10,  3.60it/s]
Epoch [1/100], Training Loss 1: 4.4810, Training Loss 2: 23653.9185, Validation Loss 1: 3.4438, Validation Loss 2: 11024.5113, Validation Accuracy: 43.00%
38it [00:10,  3.59it/s]
Epoch [2/100], Training Loss 1: 2.9458, Training Loss 2: 12478.8715, Validation Loss 1: 0.9490, Validation Loss 2: 16354.9773, Validation Accuracy: 50.00%
38it [00:10,  3.60it/s]
Epoch [3/100], Training Loss 1: 0.6610, Training Loss 2: 10153.9975, Validation Loss 1: 0.3406, Validation Loss 2: 7723.0419, Validation Accuracy: 84.00%
38it [00:10,  3.60it/s]
Epoch [4/100], Training Loss 1: 0.2150, Training Loss 2: 4067.3105, Validation Loss 1: 0.0823, Validation Loss 2: 2895.6197, Validation Accuracy: 99.00%
38it [00:10,  3.60it/s]
Epoch [5/100], Training Loss 1: 0.1054, Training Loss 2: 3098.0841, Validation Loss 1: 0.0378, Validation Loss 2: 1511.1347, Validation Accuracy: 99.00%
38it [00:10,  3.60it/s]
Epoch [6/100], Training Loss 1: 0.0310, Training Loss 2: 1620.6888, Validation Loss 1: 0.0414, Validation Loss 2: 1557.8277, Validation Accuracy: 99.33%
38it [00:10,  3.60it/s]
Epoch [7/100], Training Loss 1: 0.0220, Training Loss 2: 1337.2084, Validation Loss 1: 0.0081, Validation Loss 2: 1264.5982, Validation Accuracy: 100.00%
38it [00:10,  3.59it/s]
Epoch [8/100], Training Loss 1: 0.0096, Training Loss 2: 804.3713, Validation Loss 1: 0.0054, Validation Loss 2: 833.0199, Validation Accuracy: 100.00%
38it [00:10,  3.60it/s]
Epoch [9/100], Training Loss 1: 0.1961, Training Loss 2: 3145.8898, Validation Loss 1: 0.3409, Validation Loss 2: 4407.6376, Validation Accuracy: 86.00%
38it [00:10,  3.59it/s]
Epoch [10/100], Training Loss 1: 0.2367, Training Loss 2: 3415.5085, Validation Loss 1: 0.0863, Validation Loss 2: 2411.7473, Validation Accuracy: 99.00%
38it [00:10,  3.60it/s]
Epoch [11/100], Training Loss 1: 0.0475, Training Loss 2: 1367.7771, Validation Loss 1: 0.0459, Validation Loss 2: 1060.6453, Validation Accuracy: 99.33%
Finished Training
```

Eagle - Approach

Second Approach - Distribution Modeling approach

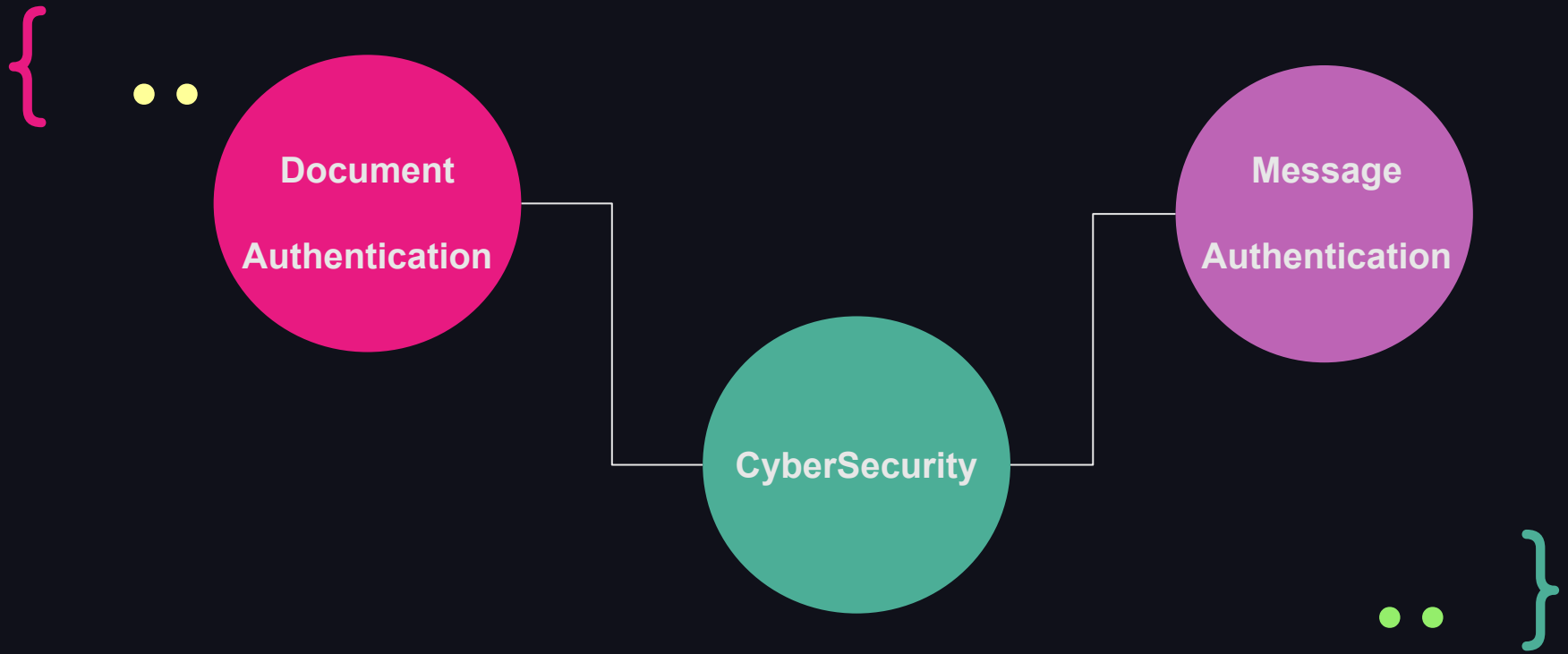
Notice:

- We noticed that provided test cases centered almost around three different distributions.
- Each distribution represents one of the three classes of messages sent (Real, Fake, Empty).
- We tried to have a simpler approach to predict class of spectrogram based on its statistical parameters.
- In this approach, we got the same accuracy as AI-Approach, but with a better inference time.

Challenges

- Resolve errors while writing the API.
- Eagle Dataset has been challenged to work with.
- CV_Hard Riddle Test case has been kind ambiguous.
- Fox Score Function is a bit challenging.

Scale Up


Document
Authentication
CyberSecurity
Message
Authentication

Scale Up

Document Authentication:

Government agencies and financial institutions can employ the module to verify the authenticity of documents containing embedded images, such as passports, IDs, or bank statements, by detecting hidden watermarks or security features.

Scale Up

Cybersecurity:

The module can be used for detecting hidden messages or steganography in images, which could be utilized by cybersecurity professionals to identify potential security threats or unauthorized communication channels.

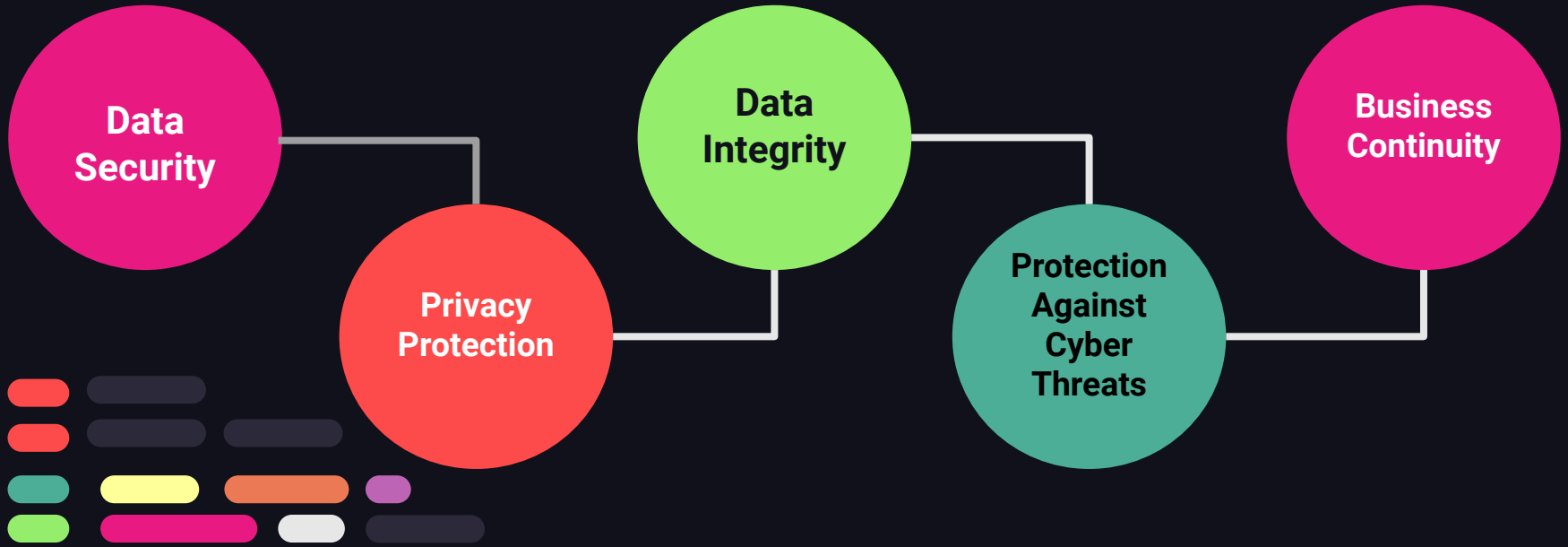
Scale Up

Message Authentication:

In secure communication protocols, verifying the integrity and authenticity of messages is crucial. The presence of hidden messages within images can serve as a form of authentication, allowing the receiver to verify that the transmitted data has not been tampered with during transmission.

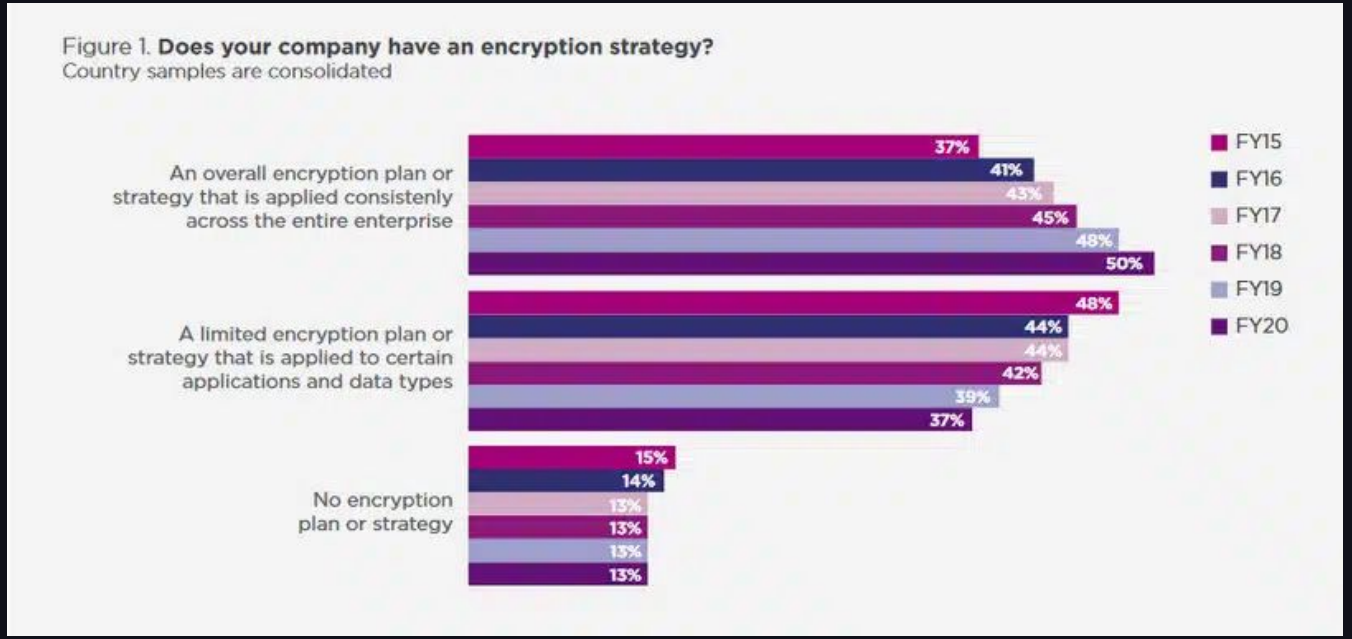
Scale Up

Benefits of using encryption



Scale Up

Benefits of using encryption



Scale Up

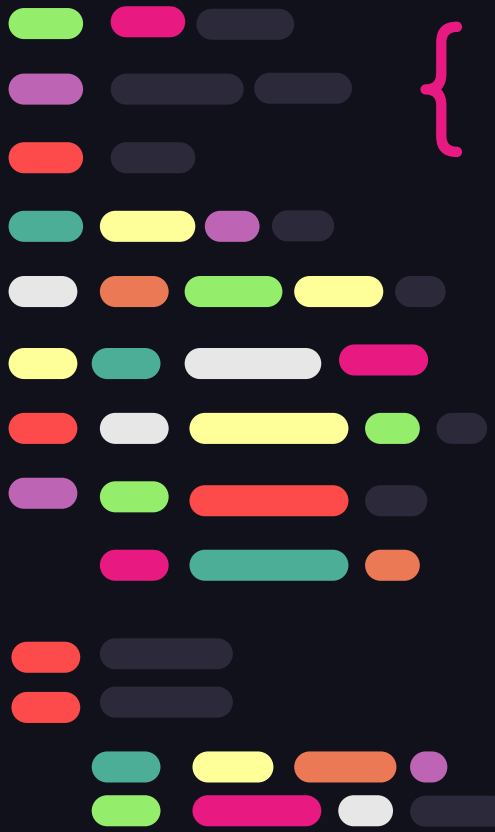
Drawbacks of using encryption

Encryption is a weapon with two sides it could be used to protect your or harming your work by encrypting your data and ask for money to decryption as we saw in may 2017 (The WannaCry ransomware attack)

It attacked many hospital and important companies lost highly sensitive data during this attack.

Possible Improvement

- Solve Computer Vision Hard Problem.
- Enhance riddles time complexity.
- Get larger dataset to have more robust model.
- Train our AI algorithm to better model noise in spectrogram and concentrate on sound features.
- Use Self Attention Network to better extract mel-spectrogram Features.



Thanks!

< Do you have any questions? >

• • •