WLMJX GMTLIV - QP TVSNIGX

Qmgleip Epjers
Mdekqe Epsrws
Nmqqc Hmppsr

SHIFT CIPHER - ML PROJECT

Michael Alfano
Izagma Alonso
Jimmy Dillon

Unit Plan by the Numbers

11-12

Grade Level

Taught after
AP CSP or AP CSA

Duration

Anywhere from
mini to full-unit

2 - 4 weeks

12

Standards

Based on CSTA
alignment

Unit Plan by Topic

How to use Shift Ciphers

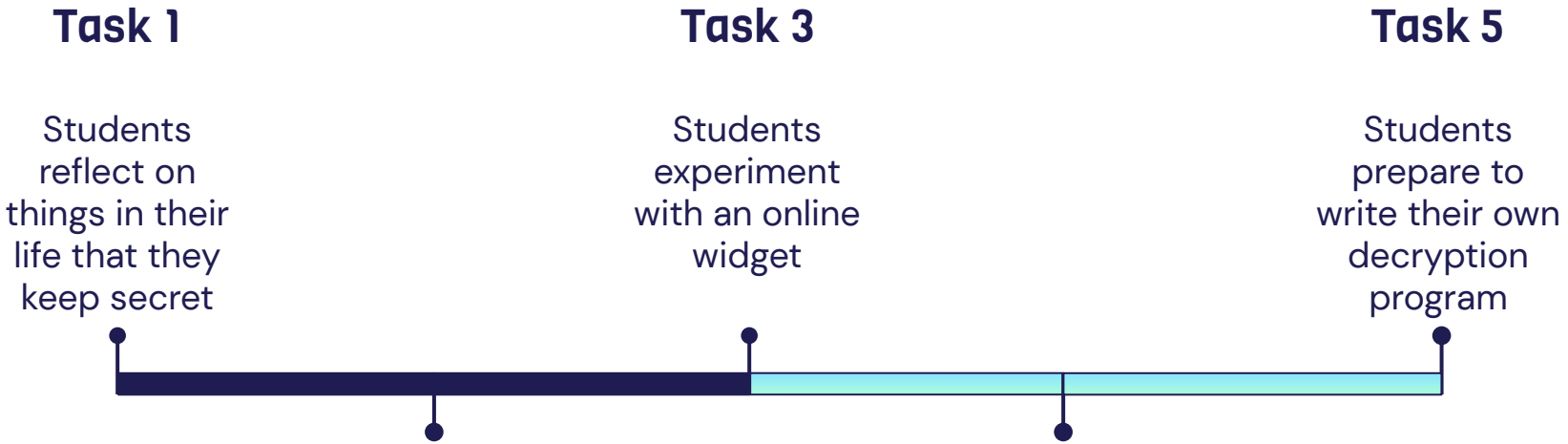
Writing a program to encode/decode using Shift Ciphers

Considerations for Machine Learning

01

SHIFT CIPHERS

Going from Unplugged to Plugged



02
WRITING A SHIFT CIPHER PROGRAM

CREATE A CODING PLAN

Step 2

Create a list of methods the program will use

Step 3

Determine data structures required

Step 1

Outline what the program needs to do

Step 4

Subgoal label each method for future use

CODING PROCESS (WRITE A PROGRAM)

Step 1

Write a single method using subgoal labels

Step 2

Test the method to ensure proper functionality

Step 3

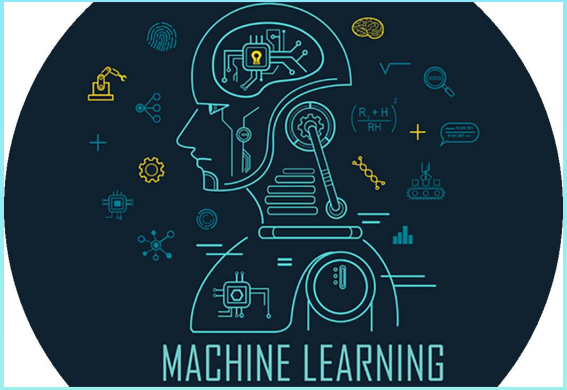
Review code written by other group members

Step 4

Run multiple methods together to test functionality

03
Machine Learning Considerations

Machine Learning (M/L)



What is Machine Learning?

Even though our code is not quite M/L, the idea of "training" the program with text can be used to introduce M/L.

What are the considerations we need to have with M/L?

Our code exemplifies the most important concepts about training data and bias which are critical for students.

What next steps should we examine in M/L?

We are becoming more reliant on predictive models and data. It is important that we integrate fairness into these models.

Class ShiftCipherNew

ATTRIBUTES

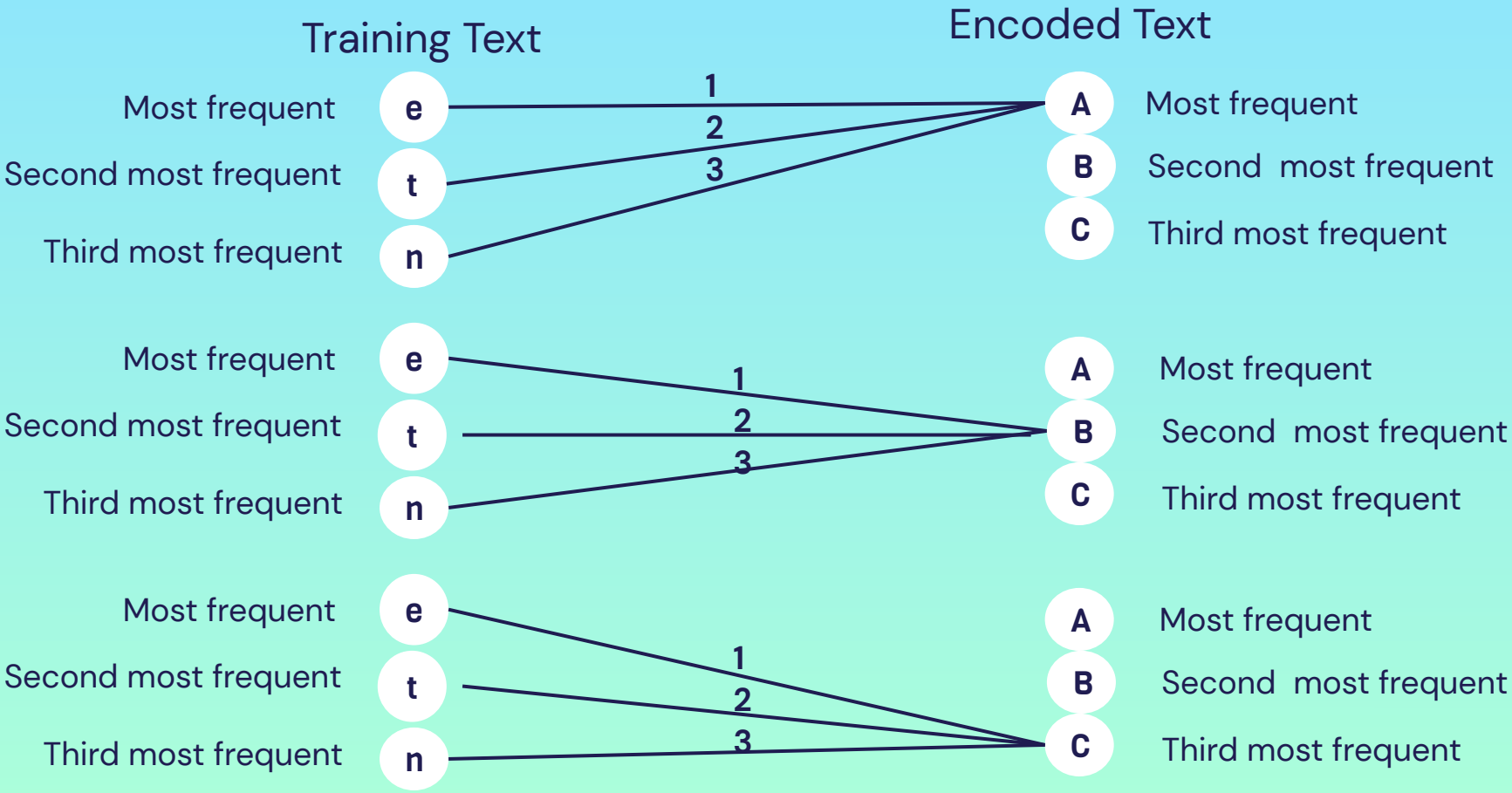
+ **String in** - encoded message provided by the user
+ **int decryptCounter** - tracks number of times decrypt () has been called
+ **int [ ] arr** - holds letter frequency data for encoded message
+ **int [ ] defaultEngFreq** - English letter frequency array if user opts not to train
+ **int [ ] engFreq** - holds letter frequency data based on user training

METHODS

+ **setInput** - Encoded message is set by the user
+ **trainFreq** - Trains the program by creating a frequency array of letters
+ **htmlToString** (unused) - Scrapes a website to train the program
+ **lowerCase** - Sets a String to all lowercase letters
+ **makeFreqArray** - Creates a letter frequency array based on a string
+ **freqLetter** - Returns the most, second most, third most… frequent letter in String
+ **shift** - Shifts a letter by a specified number
+ **calcShift** - Determines how many times to shift the alphabet
+ **decrypt** - Handles the UI and calling of all other methods

Decrypting Algorithm

MAXOPT = 3



Pedagogical Practices

01 Hands-on/Unplugged

Students interact with computer science ideas through manipulatives

02 Discovery-based Learning

Students learn the rules of computer applications through experimentation

03 Top-Down Design

Break down program into smaller parts, then build it back up piece by piece

04 Team/Pair Programming

Students collaborate in teams to effectively and efficiently complete a program

05 Project-Based Learning

Students learn by actively working to solve real-world problems

06 Rapid Research

Students extend their knowledge through research

XLI IRH!

THE END!