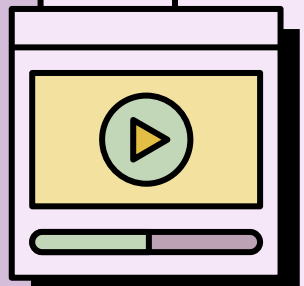
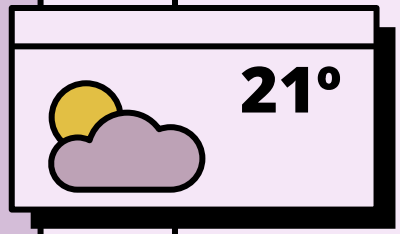
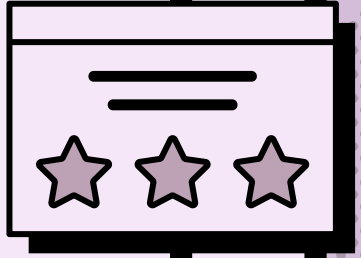
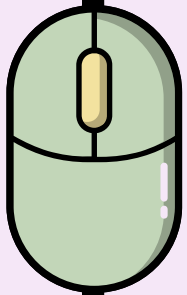
DataEthics4All









Chloe Feng with Mentor Juliana Shihadeh

Overview:

- Goal of the Project
- How it was created
- Explanation on the Project
- Results
- Demo
- What I've learned

My Project Goal:

To be able to classify the type of a building from two different classes

Terminal Towers + Flatiron Building

Dataset Overview

Using the Google Landmarks Dataset V2, curated a sub-dataset based on two buildings: Terminal Tower and the Flatiron Building.

Total amount of images
Terminal Towers: 256 images
Flatiron Building: 170 images

My Implementation Method

1. Curating a Dataset

Google Landmarks Dataset V2, Terminal Towers + Flatiron Building

2. Creating Directories

Separated the data into a training set and a testing set

3. Resizing the data

Preprocessing testing sets for both classes in for loops so images are the same size

4. Extracting RGB values

Took the RGB values from each image

5. Reshaping array

Transforming from a 3D array into a 2D array

6. Creating a model

Trained a k-means model to cluster the data into two groups

Methodology

Training + Testing accuracy

Checked how many of the predictions matched the actual label of the image. Accuracy: number of correct images/total images

Analyzing the clusters visually

I graphed the data points to see the shape of the two clusters. When the clusters overlap less it means that k-means is more accurate.

Results

Balanced Dataset = Equal # of Images in Each Class
Unbalanced Dataset = One Class had more images

The unbalanced dataset has a higher accuracy percentage the the balanced as it has more images to train on.

	Unbalanced	Balanced
Training Accuracy	75.3%	74.1%
Testing Accuracy	74.6%	72.9%

Unbalanced dataset

```
correct = 0
for i in range(340):
  if train_predictions[i] == true_labels[i]:
    correct += 1
print("Total correct value:", correct)
accuracy = correct/len(train_predictions)*100
accuracy = round(accuracy,3)
print("Training Accuracy:", accuracy,"%")
```

```
Total correct value: 256
Training Accuracy: 75.294 %
```

```
correct = 0
for i in range(85):
  if test_predictions[i] == true_labels[i]:
    correct += 1
print("Total correct value:", correct)
accuracy = correct/len(test_predictions)*100
accuracy = round(accuracy,3)
print("Testing Accuracy:", accuracy,"%")
```

```
Total correct value: 63
Testing Accuracy: 74.118 %
```

Balanced dataset

```
correct = 0
for i in range(272):
  if train_predictions[i] == true_labels[i]:
    correct += 1
print("Total correct value:", correct)
accuracy = correct/len(train_predictions)*100
accuracy = round(accuracy,3)
print("Training Accuracy:", accuracy,"%")
```

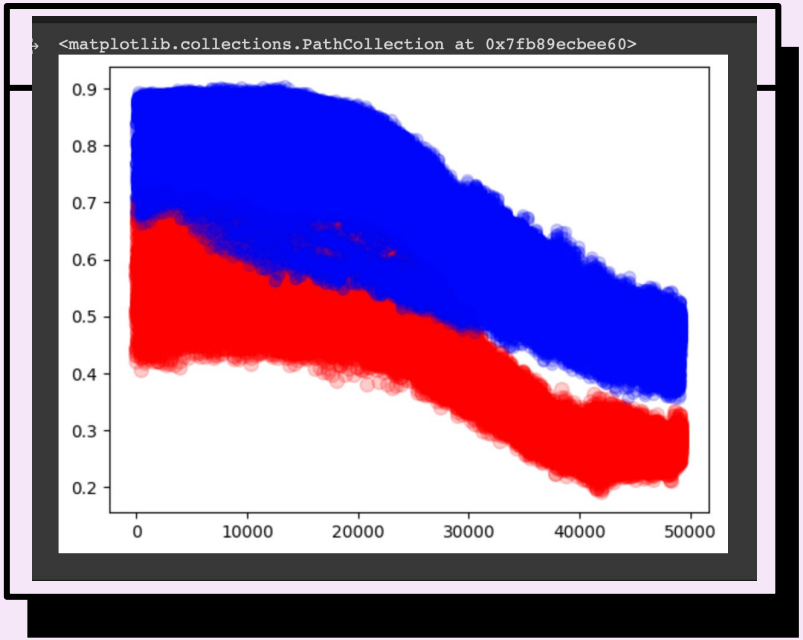
```
Total correct value: 203
Accuracy: 74.632 %
```

```
correct = 0
for i in range(85):
  if test_predictions[i] == true_labels[i]:
    correct += 1
print("Total correct value:", correct)
accuracy = correct/len(test_predictions)*100
accuracy = round(accuracy,3)
print("Testing Accuracy:", accuracy,"%")
```

```
Total correct value: 62
Testing Accuracy: 72.941 %
```

Results

An Example of a Graph of the Clusters



The datapoints partially overlap in the center

Demo

The model is able to classify whether the image is from Terminal Tower or from Flatiron Building.

My Experience

Challenges:
Learning about the basics of Python and how to effectively use different types of functions

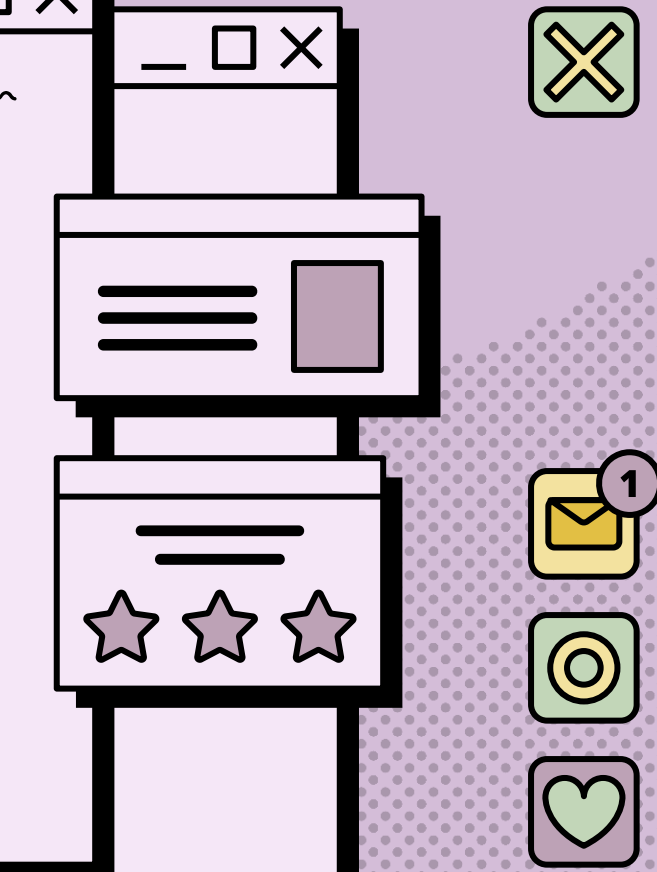
Favorites:
I enjoyed learning about the different applications of AI and also about how to code in Python.

What I've Learned

- Understanding the impacts of AI and how it works
- Different types of libraries (Machine Learning Libraries)
- Experimenting with Python, Jupyter notebook, and Google Collab

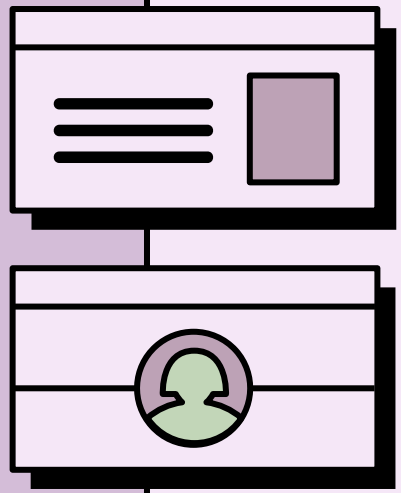
Sources:

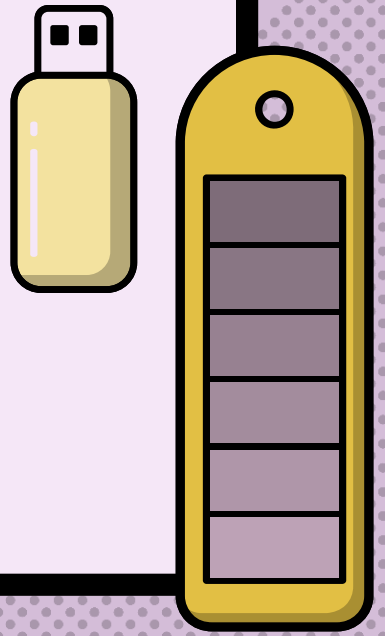
- https://storage.googleapis.com/gld-v2/web/index.html
- https://machinelearningmastery.com/save-load-machine-learning-models-python-scikit-learn/
- https://www.learndatasci.com/solutions/python-move-file/
- https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
- https://medium.com/analytics-vidhya/how-to-calculate-rgb-values-for-some-images-in-python-ccf9abcea8f3



Thanks!

-





CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

Please keep this slide for attribution