Diagram

Description automatically generated

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Introduction to Supervised Machine Learning Activity

Description

In this exercise, you will learn about the three components of an artificial intelligence (AI) system.

Instructions

1. Go to <https://drive.google.com/drive/u/1/folders/1PGO3lu_4hztRAyCPK4g8AVQtk79A79j3>, download the cat-dog data to your computer.
2. Go to: <https://teachablemachine.withgoogle.com/>
3. Click “Get Started 🡪 Image Project 🡪 Standard image model”

**Identify the three parts of an AI system in the teachable machine as discussed in class:**

Diagram

Description automatically generated with medium confidence

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

**Steps to add training dataset:**

1. Click “Class 1 🡪 Upload 🡪 Choose images from your files…”
2. Locate the training dataset in your hard drive.
3. Click “dog” and select all images. Click open.
4. Repeat with cat images for Class 2.

**Train Model**

**Add test data**

What is the output?

What happens when you select a test image other than cat and dog?

What happens if you only train on one class (say dog)?

What happens if you decrease the number of images in the training dataset?

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Machine Learning with Python Activity

Description

In this exercise, you will learn how to do clustering and classification using Python.

Instructions

**A. Iris Clustering and Classification**

1. Go to Google Colab (<https://colab.research.google.com/>) and create a new notebook A blue rectangle with white text

   AI-generated content may be incorrect.
2. Follow the tutorial in slides to build k-means clustering model and logistic regression model on the Iris data.
   1. What results do you have?

**B. Animal Image Classification**

1. Click the shared link (https://drive.google.com/drive/folders/1wCgMIwlguegGmxBvhbxdcJE1vRHeOwPT?usp=drive\_link) and sign in to your Google Drive.
2. Click the folder name animal-img, then go to **Organize** → **Add shortcut**, and add the shortcut to **My Drive**.

A screenshot of a computer

AI-generated content may be incorrect.

1. Build the image classifier:
   1. Go to Google Colab (<https://colab.research.google.com/>) and create a new notebook.
   2. Import required libraries

import tensorflow as tf

from tensorflow.keras import layers, models

import matplotlib.pyplot as plt

import numpy as np

import seaborn as sns

from sklearn.metrics import confusion\_matrix, ConfusionMatrixDisplay

* 1. Mount Google Drive and load the datasets

from google.colab import drive

drive.mount('/content/drive')

train\_dir = '/content/drive/MyDrive/animal-img/train'

test\_dir  = '/content/drive/MyDrive/animal-img/test'

IMG\_SIZE = (128, 128)

BATCH\_SIZE = 32

train\_ds = tf.keras.preprocessing.image\_dataset\_from\_directory(

    train\_dir,

    image\_size=IMG\_SIZE,

    batch\_size=BATCH\_SIZE

)

test\_ds = tf.keras.preprocessing.image\_dataset\_from\_directory(

    test\_dir,

    image\_size=IMG\_SIZE,

    batch\_size=BATCH\_SIZE

)

class\_names = train\_ds.class\_names

print("Detected Classes:", class\_names)

* 1. Build the model

base\_model = tf.keras.applications.MobileNetV2(

    input\_shape=IMG\_SIZE + (3,),

    include\_top=False,

    weights='imagenet'

)

base\_model.trainable = False  # Freeze for now

model = tf.keras.Sequential([

    layers.Rescaling(1./255, input\_shape=IMG\_SIZE + (3,)),

    base\_model,

    layers.GlobalAveragePooling2D(),

    layers.Dropout(0.3),

    layers.Dense(128, activation='relu'),

    layers.Dropout(0.3),

    layers.Dense(len(class\_names), activation='softmax')

])

* 1. Set optimizer and loss function

model.compile(optimizer='adam',

              loss='sparse\_categorical\_crossentropy',

              metrics=['accuracy'])

* 1. Train the model

model.compile(optimizer='adam',

              loss='sparse\_categorical\_crossentropy',

              metrics=['accuracy'])

1. Test the model

test\_images = []

test\_labels = []

for image, label in test\_ds.unbatch():

    test\_images.append(image)

    test\_labels.append(label)

test\_images = np.array(test\_images)

test\_labels = np.array(test\_labels)

# Prompt user for a class name

print("Available classes:", class\_names)

selected\_class = input("Enter a class name from the list above: ").strip()

# Get class index and filter images

class\_index = class\_names.index(selected\_class)

indices = np.where(test\_labels == class\_index)[0]

# Predict labels for all selected images

predicted\_labels = []

for idx in indices:

    img = tf.expand\_dims(test\_images[idx], axis=0)

    pred = model.predict(img, verbose=0)

    predicted\_labels.append(np.argmax(pred))

# Plot the images

plt.figure(figsize=(15, 8))

for i, idx in enumerate(indices):

    plt.subplot(3, int(np.ceil(len(indices) / 3)), i + 1)

    plt.imshow(test\_images[idx].astype("uint8"))

    plt.axis('off')

    true\_label = class\_names[test\_labels[idx]]

    pred\_label = class\_names[predicted\_labels[i]]

    plt.title(f"T: {true\_label}\nP: {pred\_label}", fontsize=9)

plt.tight\_layout()

plt.show()

* 1. Which classes have lower prediction accuracy?
  2. Find the misclassified images in the test data. Can you guess why they were incorrectly predicted?

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

### Can AI Really Code?

Description

In this exercise, you will ask ChatGPT to write simple Python code.

Instructions

You can work in group or individually and ask ChatGPT to write simple Python programs to solve following problems:

1. Convert temperatures from Fahrenheit to Celsius.
2. Calculates the sum of numbers from 1 to 100.
3. Create a text-based adventure game.