

[ CAT vs DOG Prediction using ML ]

```
In [190... import numpy as np
import random
import matplotlib.pyplot as plt
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dense, Flatten
```

Load dataset

```
In [193... X_train =np.loadtxt('input.csv', delimiter=',')
Y_train =np.loadtxt('labels.csv', delimiter=',')

X_test =np.loadtxt('input_test.csv', delimiter=',')
Y_test =np.loadtxt('labels_test.csv', delimiter=',')
```

```
In [194... X_train =X_train.reshape(len(X_train), 100, 100, 3)
Y_train=Y_train.reshape(len(Y_train), 1)

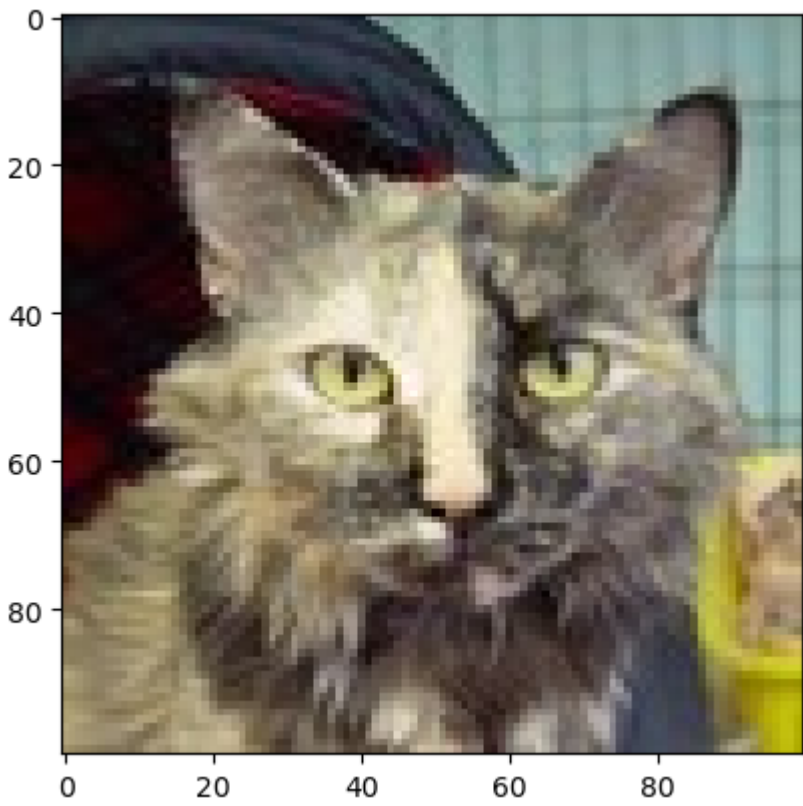
X_test =X_test.reshape(len(X_test), 100, 100, 3)
Y_test=Y_test.reshape(len(Y_test), 1)

X_train = X_train/255.0
X_test = X_test/255.0
```

```
In [195... X_train.shape,Y_train.shape,X_test.shape,Y_test.shape
```

Out[195... ((2000, 100, 100, 3), (2000, 1), (400, 100, 100, 3), (400, 1))

```
In [196... idx =random.randint(0, len(X_train))
plt.imshow(X_train[idx, :])
plt.show()
```



Model

```
In [198... model =Sequential([
    Conv2D(23, (3,3),activation = 'relu', input_shape =(100,100,3)),
    MaxPooling2D((2,2)),

    Conv2D(32, (3,3),activation = 'relu'),
    MaxPooling2D((2,2)),

    Flatten(),
    Dense(64,activation = 'relu'),
    Dense(1,activation = 'sigmoid'),

])
```

```
In [199... model =Sequential()
model.add(Conv2D(23, (3,3),activation = 'relu', input_shape =(100,100,3)))
model.add(MaxPooling2D((2,2)))

model.add(Conv2D(23, (3,3),activation = 'relu',))
model.add(MaxPooling2D((2,2)))

model.add(Flatten())
model.add(Dense(64,activation = 'relu'))
model.add(Dense(1,activation = 'sigmoid'))
```

```
In [200... # opt=keras.optimizers.SGD(learning_rate =0.001)
model.compile(loss = 'binary_crossentropy', optimizer='adam', metrics = ['accuracy'])
```

```
In [201... model.fit(X_train, Y_train,epochs =5 ,batch_size =64)
```

Epoch 1/5  
32/32 ----- 4s 90ms/step - accuracy: 0.5525 - loss: 0.6842  
Epoch 2/5  
32/32 ----- 3s 89ms/step - accuracy: 0.6170 - loss: 0.6432  
Epoch 3/5  
32/32 ----- 3s 87ms/step - accuracy: 0.7030 - loss: 0.5813  
Epoch 4/5  
32/32 ----- 3s 90ms/step - accuracy: 0.7260 - loss: 0.5377  
Epoch 5/5  
32/32 ----- 3s 84ms/step - accuracy: 0.7830 - loss: 0.4666

Out[201... <keras.src.callbacks.history.History at 0x22f346dbaa0>

```
In [202... model.evaluate(X_test, Y_test)
```

13/13 ----- 0s 14ms/step - accuracy: 0.6600 - loss: 0.6114

Out[202... [0.6113585829734802, 0.6600000262260437]

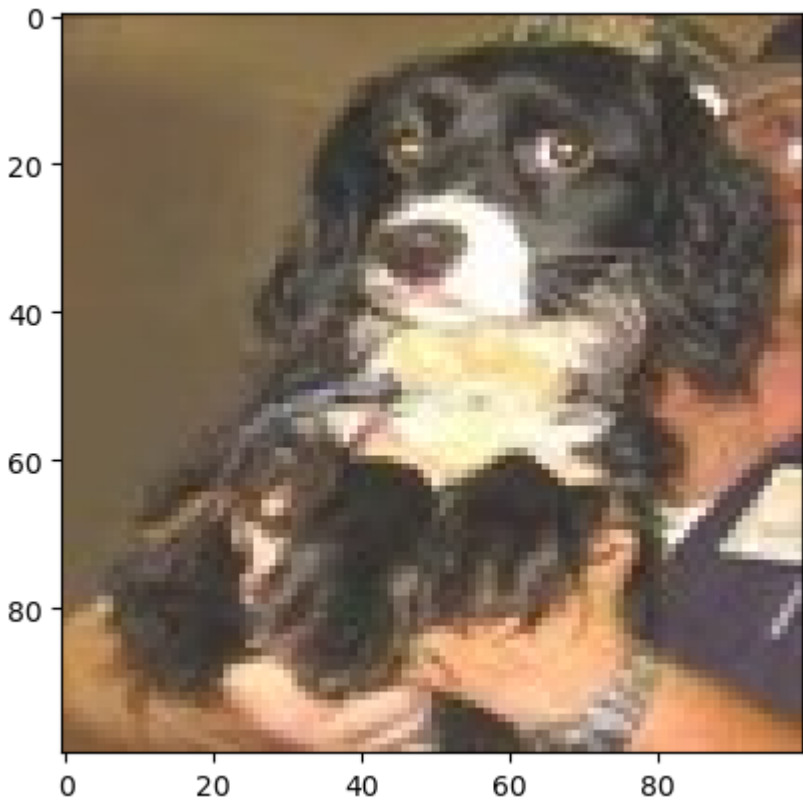
Making predictions

```
In [226... idx2 =random.randint(0, len(Y_test))
plt.imshow(X_test[idx2, :])
plt.show()

y_pred =model.predict(X_test[idx2, :].reshape(1, 100, 100, 3))
y_pred =y_pred>0.5

if(y_pred ==0):
    y_pred = 'dog'
else:
    y_pred = 'cat'

print("Our Model says it is a:", y_pred)
```



1/1 ----- 0s 52ms/step  
Our Model says it is a: dog

**\*\* END - PROJECT \*\***