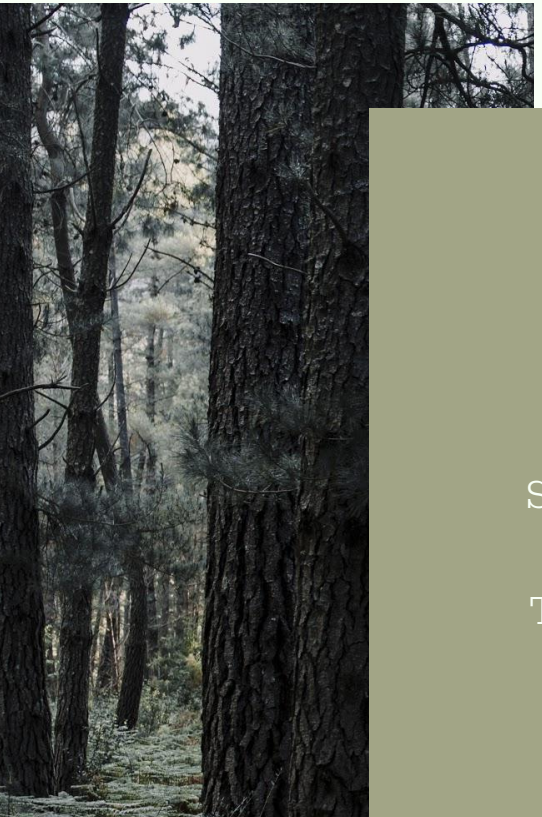

Deep Foraging
by Nick Sherwin
edible

Objective

To accurately classify whether a wild mushroom is edible or not.

poisonous

Methodology

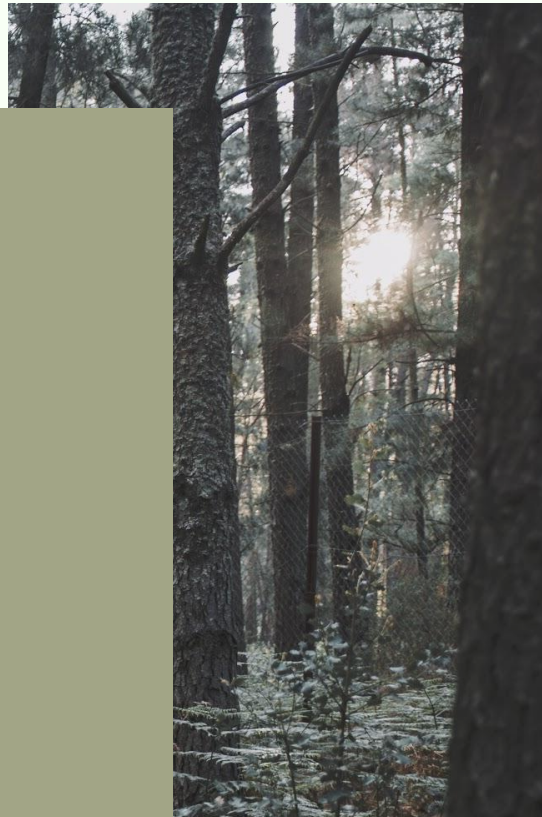


Tools

SciKit Learn
Keras
TensorFlow

Data

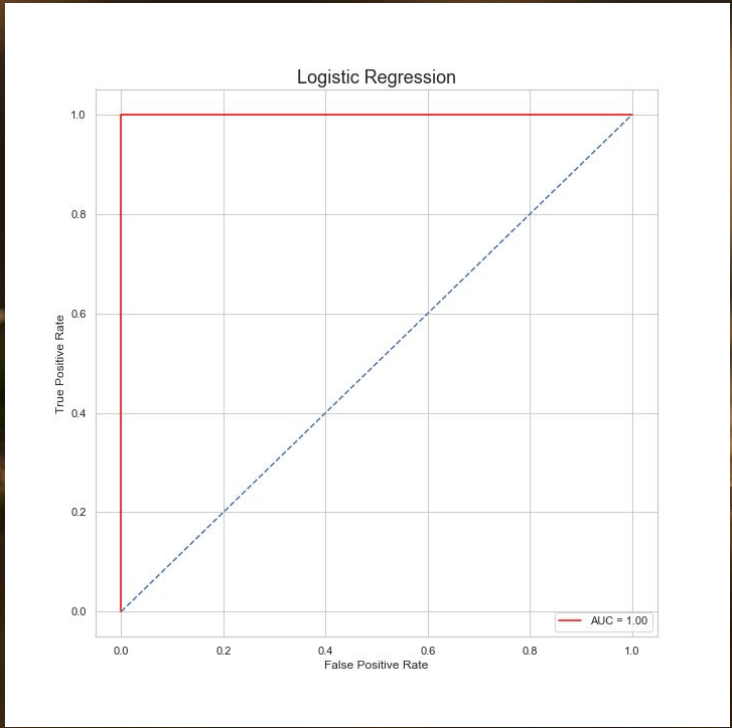
Kaggle
UCI
ImageNet



Part I. Classification Process

- Define Models + Metrics
- Encode + Dummify
- Interpret Coefficients

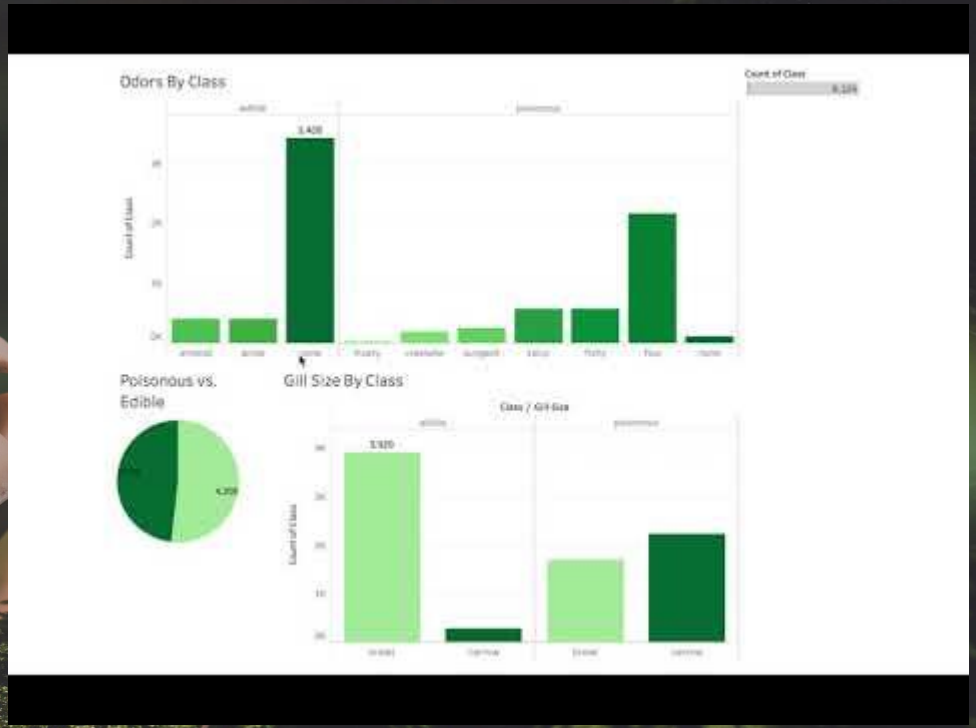
Part I. Logistic Regression Results



- Accuracy Score: 99.8%
- Precision + Recall + F1: 1.0
- 1 False Negative
- oh...but the test was 100% on everything

edible

Part I. Feature Importance



poisonous

Part II. Image Classification Process

- Batch Download Images
- CNNs + Transfer Learning
- Iterate

edible

Part II. Image Classification Results

```
INFO:tensorflow:2019-08-04 14:09:14.062468: Step 3970: Validation accuracy = 71.0% (N=100)
INFO:tensorflow:2019-08-04 14:09:14.692205: Step 3980: Train accuracy = 92.0%
INFO:tensorflow:2019-08-04 14:09:14.692328: Step 3980: Cross entropy = 0.274503
INFO:tensorflow:2019-08-04 14:09:14.753056: Step 3980: Validation accuracy = 72.0% (N=100)
INFO:tensorflow:2019-08-04 14:09:15.383123: Step 3990: Train accuracy = 88.0%
INFO:tensorflow:2019-08-04 14:09:15.383255: Step 3990: Cross entropy = 0.341200
INFO:tensorflow:2019-08-04 14:09:15.441476: Step 3990: Validation accuracy = 66.0% (N=100)
INFO:tensorflow:2019-08-04 14:09:15.998436: Step 3999: Train accuracy = 93.0%
INFO:tensorflow:2019-08-04 14:09:15.[illegible]: Step 3999: Cross entropy = 0.275459
INFO:tensorflow:2019-08-04 14:09:1[illegible]: Step 3999: Validation accuracy = 68.0% (N=100)
INFO:tensorflow:Final test accura[illegible] = 72.4% (N=[illegible]23)
```

```
- 992s 124ms/step - loss: 0.1587 - acc: 0.9344
```

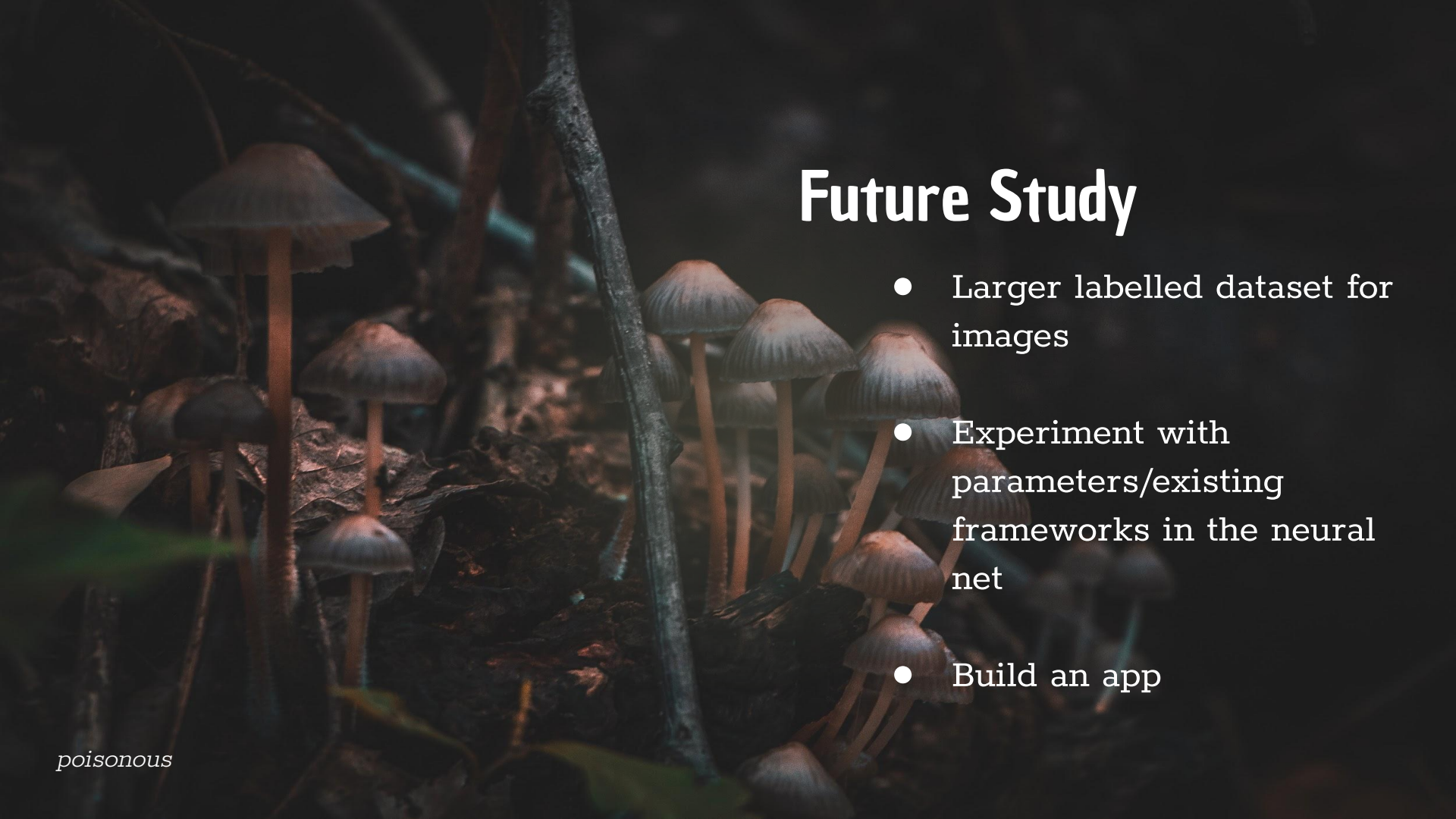
```
INFO:tensorflow:2019-08-05 12:04:13.365216: Step 3980: Train accuracy = 100.0%
INFO:tensorflow:2019-08-05 12:04:13.365347: Step 3980: Cross entropy = 0.051117
INFO:tensorflow:2019-08-05 12:04:13.422062: Step 3980: Validation accuracy = 86.0% (N=100)
INFO:tensorflow:2019-08-05 12:04:14.013232: Step 3990: Train accuracy = 99.0%
INFO:tensorflow:2019-08-05 12:04:14.013363: Step 3990: Cross entropy = 0.076969
INFO:tensorflow:2019-08-05 12:04:14.071646: Step 3990: Validation accuracy = 91.0% (N=100)
INFO:tensorflow:2019-08-05 12:04:14.596354: Step 3999: Train accuracy = 98.0%
INFO:tensorflow:2019-08-05 12:04:14.[illegible]: Step 3999: Cross entropy = 0.094274
INFO:tensorflow:2019-08-05 12:04:1[illegible]: Step 3999: Validation accuracy = 89.0% (N=100)
INFO:tensorflow:Final test accura[illegible] = 89.6% (N=[illegible]2)
```

mislabled as edible. this is a death cap. it will kill you.

Recommendations

- Sniff your shrooms
- Measure your gills
- Consult an expert before consuming

poisonous

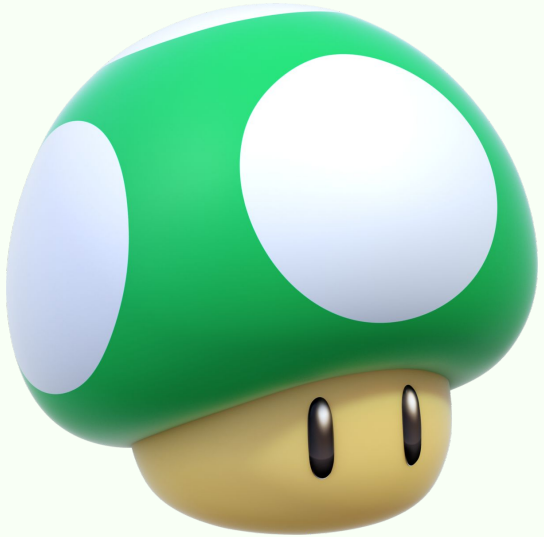


Future Study

- Larger labelled dataset for images
- Experiment with parameters/existing frameworks in the neural net
- Build an app

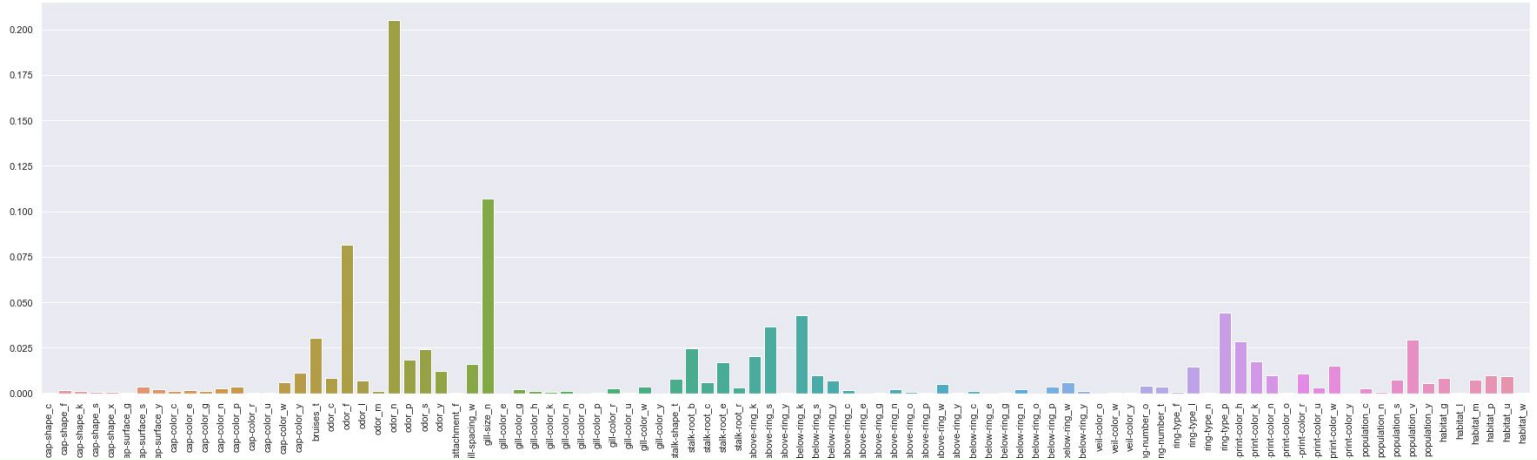
poisonous

Thank you.



edible and gives 1-UP

Appendix: Features



Naive Bayes

Test results:

Accuracy Score: 0.9502

Classification Report:

	precision	recall	f1-score	support
0	1.00	0.91	0.95	845
1	0.91	1.00	0.95	780
micro avg	0.95	0.95	0.95	1625
macro avg	0.95	0.95	0.95	1625
weighted avg	0.95	0.95	0.95	1625

Confusion Matrix:

```
[[767  78]
 [  3 777]]
```

SVC

Test results:

Accuracy Score: 0.9963

Classification Report:

	precision	recall	f1-score	support
0	0.99	1.00	1.00	845
1	1.00	0.99	1.00	780
micro avg	1.00	1.00	1.00	1625
macro avg	1.00	1.00	1.00	1625
weighted avg	1.00	1.00	1.00	1625

Confusion Matrix:

```
[[845   0]
 [  6 774]]
```

Random Forest

Test results:

Accuracy Score: 0.9580

Classification Report:

	precision	recall	f1-score	support
0	0.96	0.92	0.94	53
1	0.96	0.98	0.97	90
micro avg	0.96	0.96	0.96	143
macro avg	0.96	0.95	0.95	143
weighted avg	0.96	0.96	0.96	143

Confusion Matrix:

[[49 4]
 [2 88]]

CNN Code

In [2]:
```
# Initialising the CNN
classifier = Sequential()
```

In [3]:
```
# Step 1 - Convolution
classifier.add(Conv2D(32, (3, 3), input_shape = (64, 64, 3), activation = 'relu'))
```

In [4]:
```
# Step 2 - Pooling
classifier.add(MaxPooling2D(pool_size = (2, 2)))
```

In [5]:
```
# Adding a second convolutional layer
classifier.add(Conv2D(32, (3, 3), activation = 'relu'))
classifier.add(MaxPooling2D(pool_size = (2, 2)))
```

In [6]:
```
# Step 3 - Flattening
classifier.add(Flatten())
```

In [7]:
```
# Step 4 - Full connection
classifier.add(Dense(units = 128, activation = 'relu'))
classifier.add(Dense(units = 1, activation = 'sigmoid'))
```

In [8]:
```
# Compiling the CNN
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
```

CNN Fit

In [9]:

```
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(rescale = 1./255,
                                   shear_range = 0.2,
                                   zoom_range = 0.2,
                                   horizontal_flip = True)

test_datagen = ImageDataGenerator(rescale = 1./255)

training_set = train_datagen.flow_from_directory('mushroom_images',
                                                 target_size = (64, 64),
                                                 batch_size = 16,
                                                 class_mode = 'binary')

test_set = test_datagen.flow_from_directory('test_images',
                                            target_size = (64, 64),
                                            batch_size = 16,
                                            class_mode = 'binary')

classifier.fit_generator(training_set,
                         steps_per_epoch = 8000,
                         epochs = 10,
                         validation_data = test_set,
                         validation_steps = 2000)
```