Internet Archive PDF

Huilin Chang, Yihnew Eshetu, Celeste Lemrow

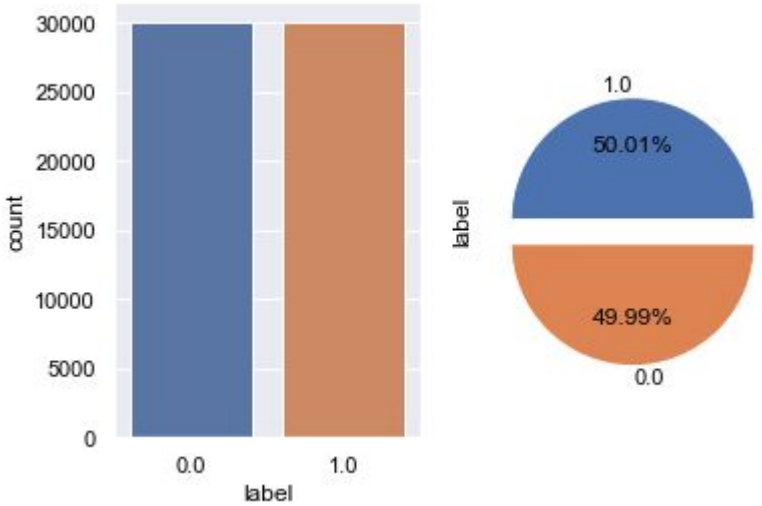
Faculty Advisor : Professor Alvarado
Client : Internet Archive
Sponsor : Bryan Newbold

Progress Made

- Balanced Data
- Data Pipeline
- Data Feature Engineering
- Models
  - Text Based Models
    - XGBoost
    - Keras
    - SVM
  - Image Based Model
    - Keras (VGG16)

Balanced Data

- Added an additional 20k non research papers



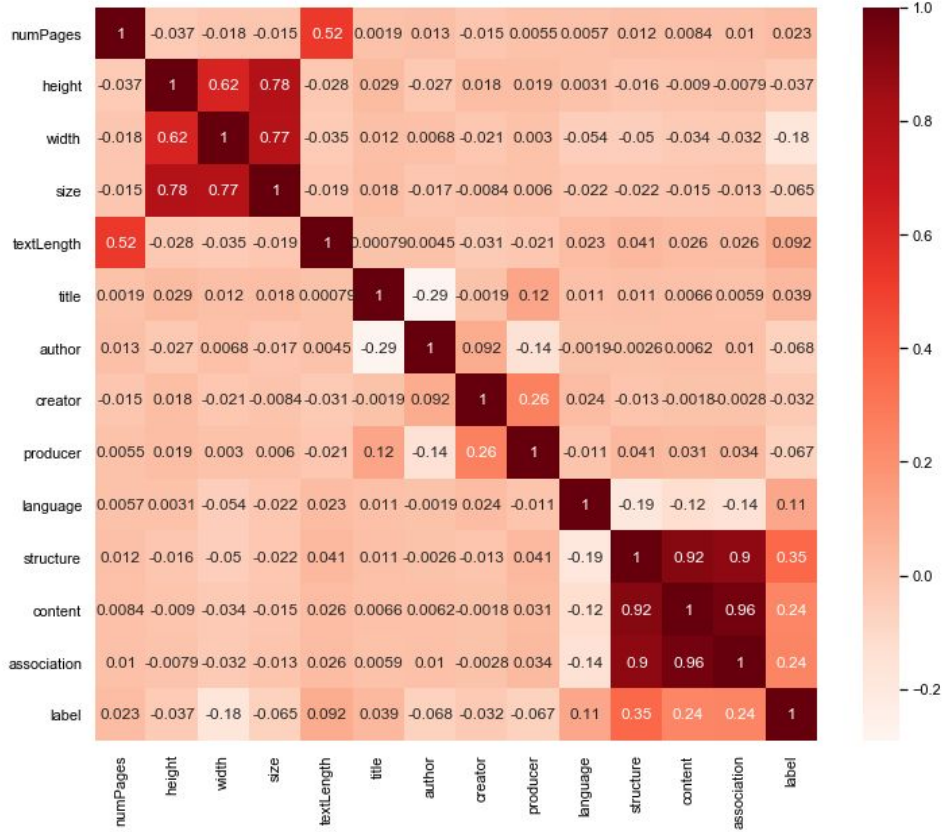
Data Pipeline

- Modified our data pipeline to use multiprocessing (40 cores)
- Restructured our images directory to include test and training subfolders

Task	**Time using 1 Processor**	**Time using Multiprocessing**
Extract meta and text	6 hours	10 minutes
Convert PDFs to images	3 hours	5 minutes

Data Feature Engineering

- The use of multiprocessing allows for further feature extraction
  - Ability to look for keywords in text
    - English
    - Non-english
      - Translate keywords to the language of the text
  - Process adds 14 minutes to the additionally extraction of meta and text data

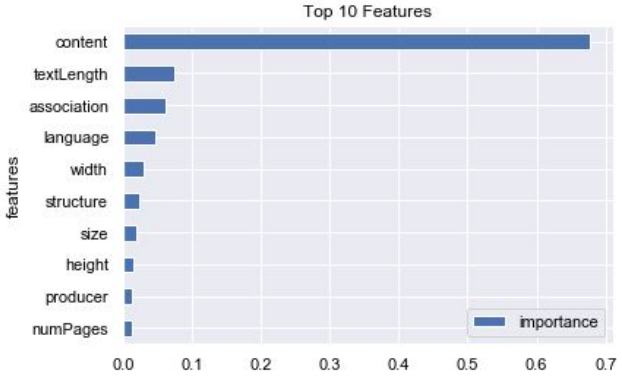


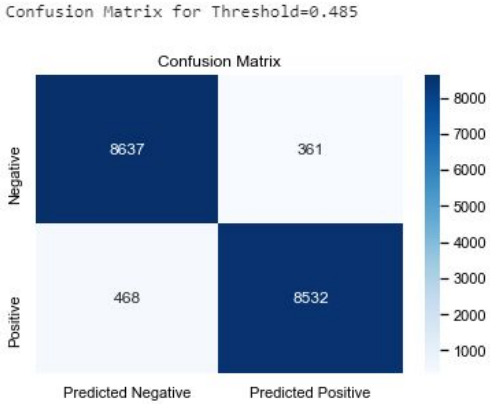
Text Based Models

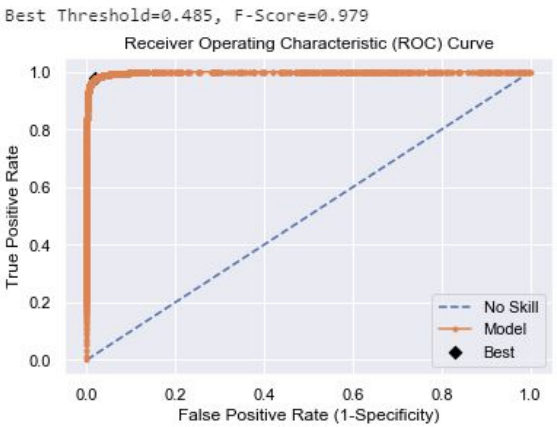
Model Type	Accuracy
XGBoost	95.39%
Keras	93.89%
SVM	90.40%

XGBoost

- Grid Search
  - N Estimators, Learning Rate, Depth
  - 3 k-fold
- F-score : 97.90%
- Accuracy: 95.39%







Keras-Tensorflow

- Model Structure
  - Input dimension of 14
  - One hidden layer
  - Adam optimizer
  - Epochs 100
- Accuracy: 93.89%

```
from keras.models import Sequential
from keras.layers import Dense
import tensorflow as tf

model = Sequential()

model.add(Dense(2048, activation='relu', input_shape=(14,)))
model.add(Dense(1024, activation='relu', ))
opt = keras.optimizers.Adam(learning_rate = 0.001)

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

```
model.compile(loss='binary_crossentropy',
              optimizer= opt,
              metrics=['accuracy'])

model.fit(X_train, y_train,epochs=100, batch_size=1, verbose=1)
```

```
Epoch 88/100
41994/41994 [==============================] - 168s 4ms/step - loss: 0.1816 - accuracy: 0.9367
Epoch 89/100
41994/41994 [==============================] - 170s 4ms/step - loss: 0.2345 - accuracy: 0.9362
Epoch 90/100
41994/41994 [==============================] - 168s 4ms/step - loss: 0.1873 - accuracy: 0.9367
Epoch 91/100
41994/41994 [==============================] - 168s 4ms/step - loss: 0.2435 - accuracy: 0.9367
Epoch 92/100
41994/41994 [==============================] - 170s 4ms/step - loss: 0.3206 - accuracy: 0.9373
Epoch 93/100
41994/41994 [==============================] - 169s 4ms/step - loss: 0.2495 - accuracy: 0.9371
Epoch 94/100
41994/41994 [==============================] - 167s 4ms/step - loss: 0.2976 - accuracy: 0.9367
Epoch 95/100
41994/41994 [==============================] - 166s 4ms/step - loss: 0.3050 - accuracy: 0.9371
Epoch 96/100
20694/41994 [============>.................] - ETA: 1:24 - loss: 0.1743 - accuracy: 0.9389
```

Image Based Model

- Leveraged an existing Keras application, VGG16, for large scale image classification

Model Type	Accuracy
Keras (VGG16)	90.01%

```
Model: "vgg16"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 256, 256, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 256, 256, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 128, 128, 64)      0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 128, 128, 128)     73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 128, 128, 128)     147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 64, 64, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 64, 64, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 64, 64, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 64, 64, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 32, 32, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 32, 32, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 32, 32, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 32, 32, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 16, 16, 512)       0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 16, 16, 512)       2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 8, 8, 512)         0
=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
```

Further Work

- Identify possible additional features
- Remove features that are insignificant to the models
- Machine learning model hyperparameter tuning
- Compare the pros and cons of text-based vs image-based models