Fashion Image Classification

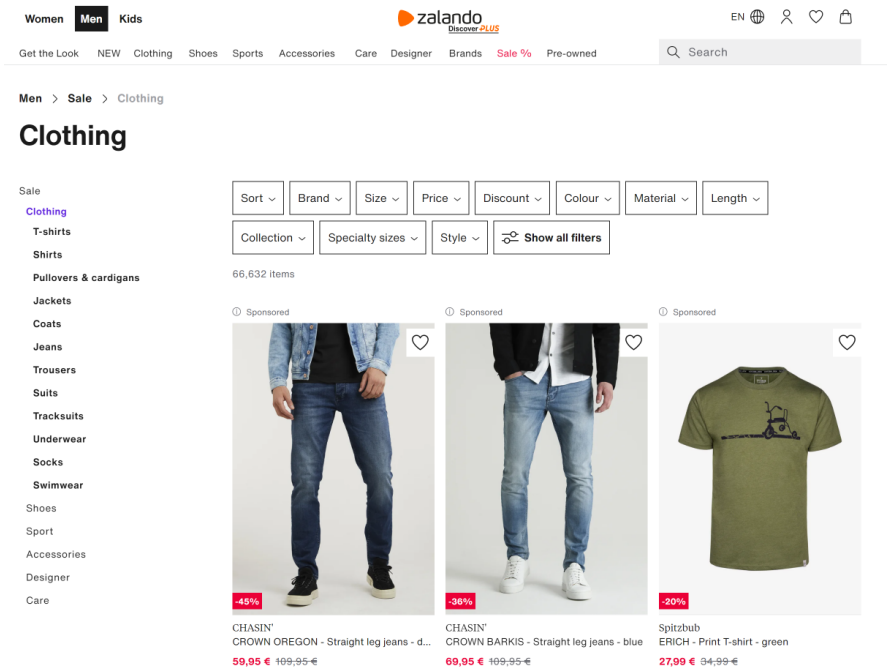
Shashank Nagaraja

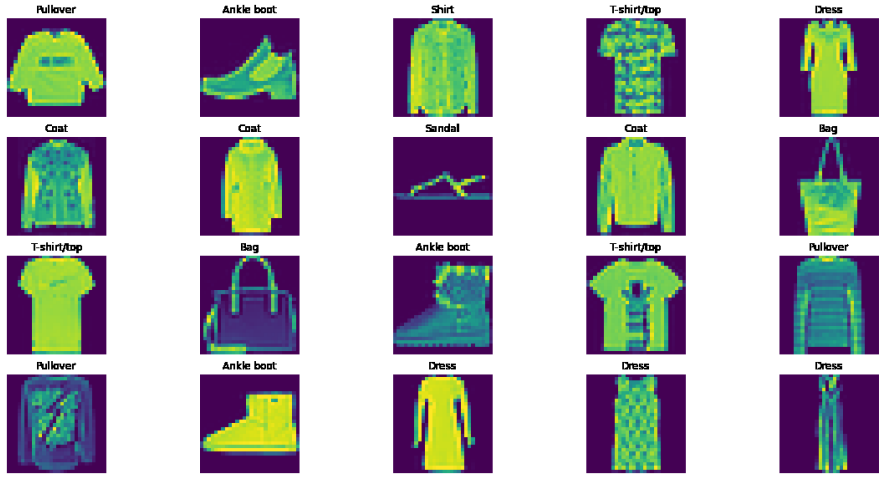
IST 707 Final Presentation

Dataset

- Zalando: German e-commerce website for clothes & accessories. Operations throughout Europe.
- Train-test structure very similar to MNIST dataset

	Train	Test
# of Images	60,000	10,000
Image Size	28 X 28 Greyscale	28 X 28 Greyscale
Classes	10	10

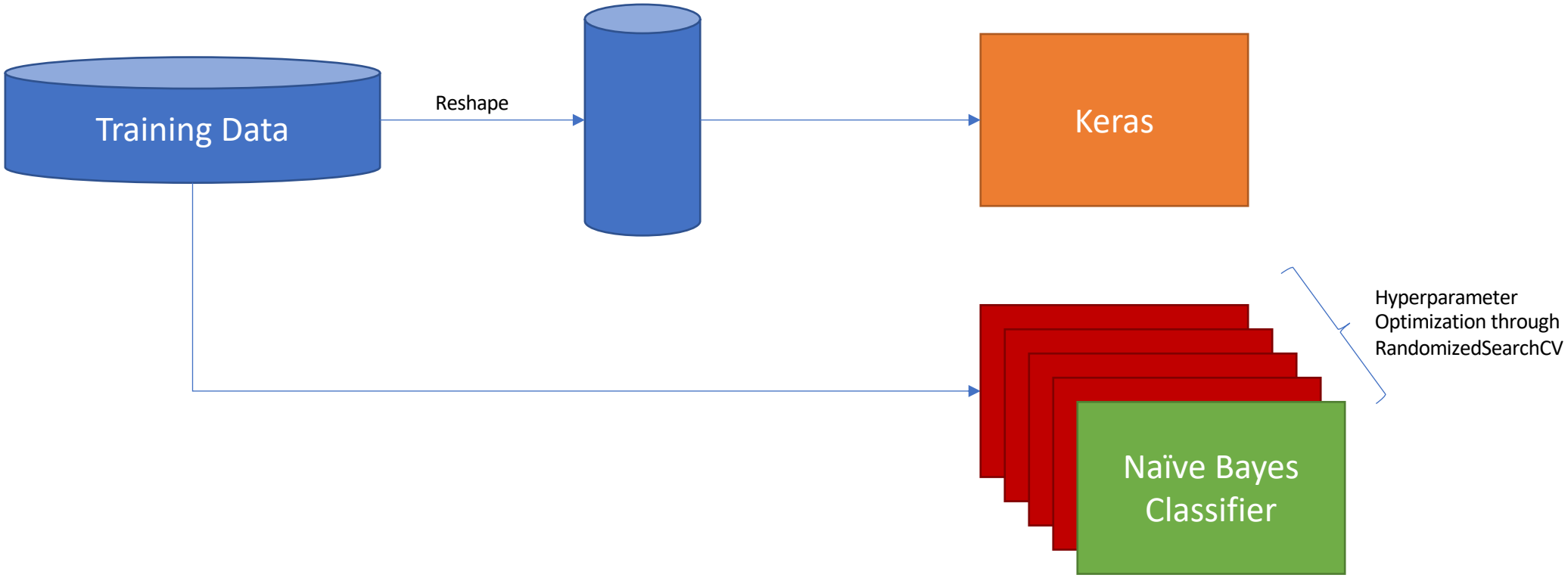




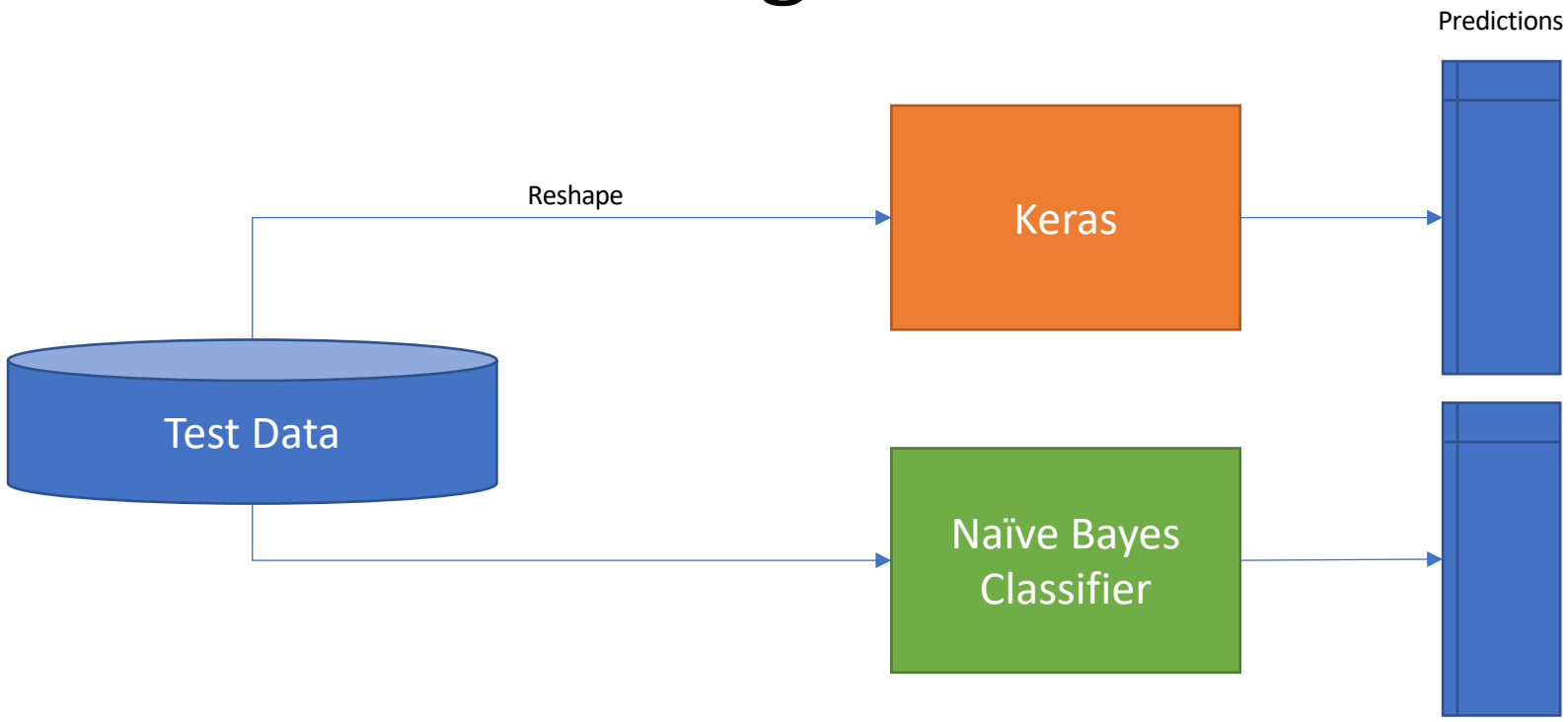
Why use Fashion MNIST?

- Conventional MNIST dataset is used to benchmark & validate classification algorithms
  - \+ Image size & shape is same across data
  - \+ Balanced classes
  - \+ Moderately sized dataset
  - \- Too easy; modern algorithms can ace MNIST
  - \- Not representative of actual Computer Vision tasks
  - \- Low complexity
  - \- 1px difference between classes

Model Building

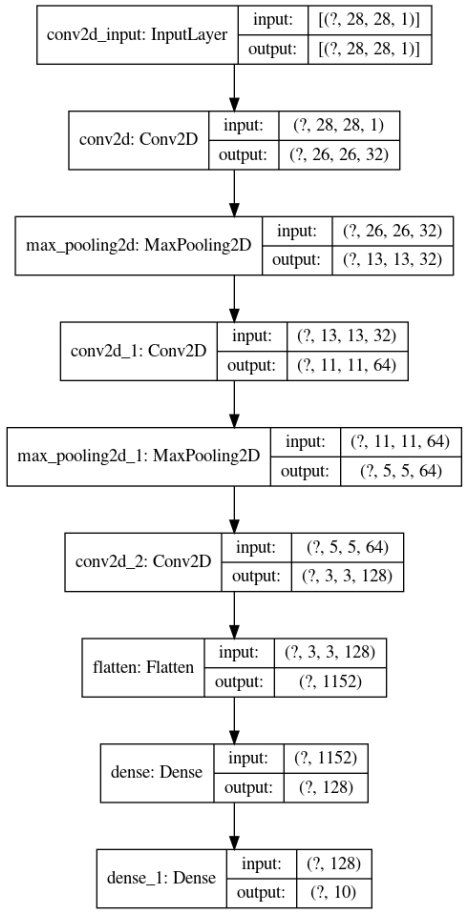


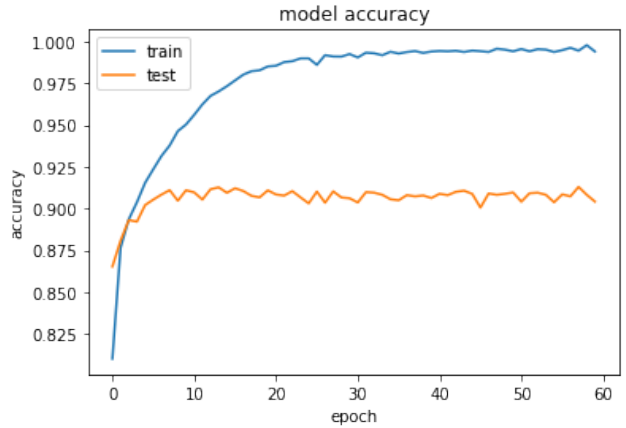
Model Building

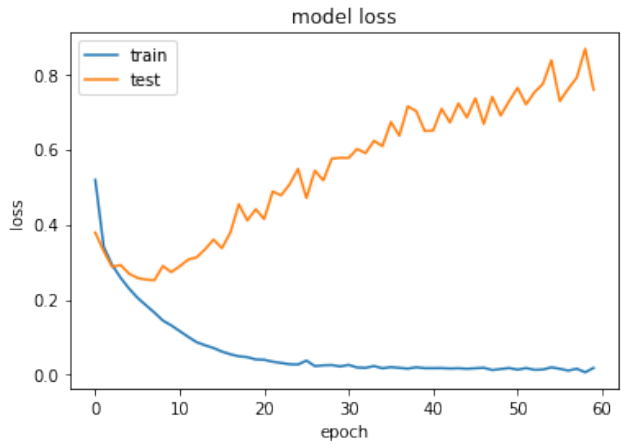


- Accuracy
- F-score
- Precision
- Recall

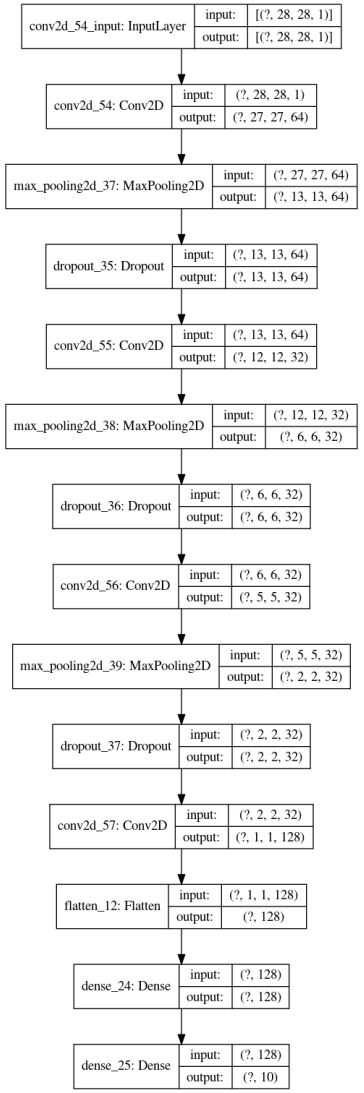
Keras Model

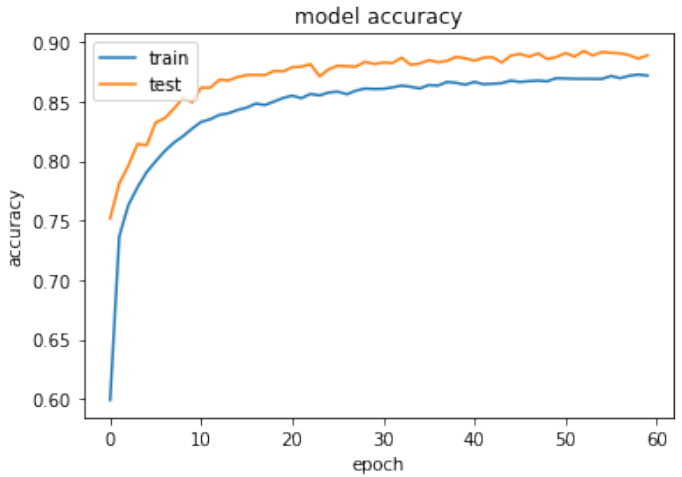

conv2d_input: InputLayer
input: [(?, 28, 28, 1)]
output: [(?, 28, 28, 1)]
conv2d: Conv2D
input: (?, 28, 28, 1)
output: (?, 26, 26, 32)
max_pooling2d: MaxPooling2D
input: (?, 26, 26, 32)
output: (?, 13, 13, 32)
conv2d_1: Conv2D
input: (?, 13, 13, 32)
output: (?, 11, 11, 64)
max_pooling2d_1: MaxPooling2D
input: (?, 11, 11, 64)
output: (?, 5, 5, 64)
conv2d_2: Conv2D
input: (?, 5, 5, 64)
output: (?, 3, 3, 128)
flatten: Flatten
input: (?, 3, 3, 128)
output: (?, 1152)
dense: Dense
input: (?, 1152)
output: (?, 128)
dense_1: Dense
input: (?, 128)
output: (?, 10)

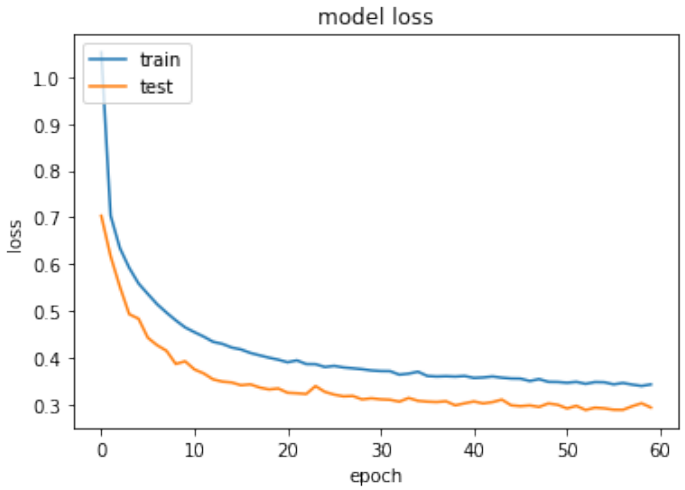

model accuracy
train
test
accuracy
epoch
1.000
0.975
0.950
0.925
0.900
0.875
0.850
0.825
0
10
20
30
40
50
60


model loss
train
test
loss
epoch
0.8
0.6
0.4
0.2
0.0
0
10
20
30
40
50
60

Keras Model

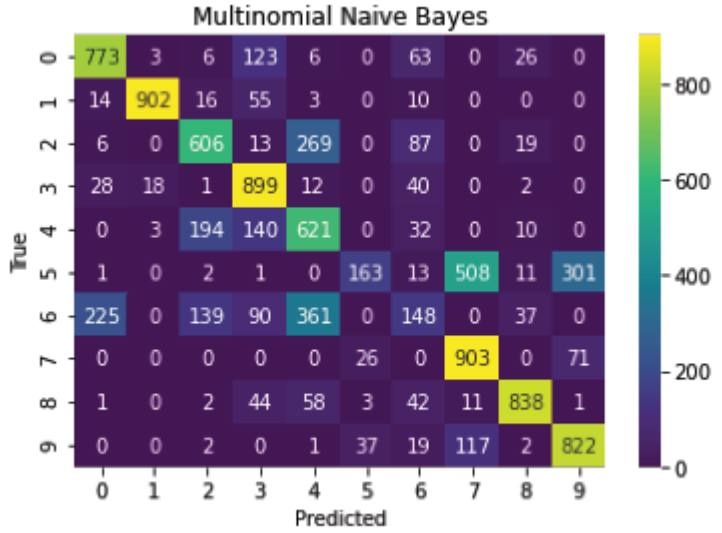

conv2d_54_input: InputLayer
input: [(?, 28, 28, 1)]
output: [(?, 28, 28, 1)]
conv2d_54: Conv2D
input: (?, 28, 28, 1)
output: (?, 27, 27, 64)
max_pooling2d_37: MaxPooling2D
input: (?, 27, 27, 64)
output: (?, 13, 13, 64)
dropout_35: Dropout
input: (?, 13, 13, 64)
output: (?, 13, 13, 64)
conv2d_55: Conv2D
input: (?, 13, 13, 64)
output: (?, 12, 12, 32)
max_pooling2d_38: MaxPooling2D
input: (?, 12, 12, 32)
output: (?, 6, 6, 32)
dropout_36: Dropout
input: (?, 6, 6, 32)
output: (?, 6, 6, 32)
conv2d_56: Conv2D
input: (?, 6, 6, 32)
output: (?, 5, 5, 32)
max_pooling2d_39: MaxPooling2D
input: (?, 5, 5, 32)
output: (?, 2, 2, 32)
dropout_37: Dropout
input: (?, 2, 2, 32)
output: (?, 2, 2, 32)
conv2d_57: Conv2D
input: (?, 2, 2, 32)
output: (?, 1, 1, 128)
flatten_12: Flatten
input: (?, 1, 1, 128)
output: (?, 128)
dense_24: Dense
input: (?, 128)
output: (?, 128)
dense_25: Dense
input: (?, 128)
output: (?, 10)


model accuracy
train
test
accuracy
epoch
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0
10
20
30
40
50
60

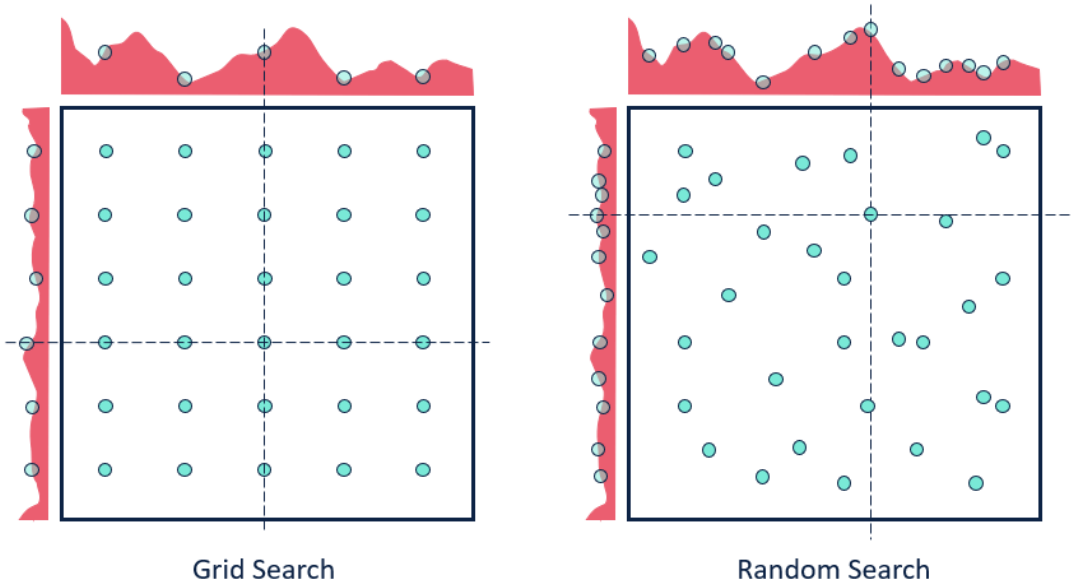

model loss
train
test
loss
epoch
1.0
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0
10
20
30
40
50
60

Naïve Bayes

- Alpha
- Prior Fit



RandomizedSearchCV



Results

- Keras Model has the highest accuracy (~ 90%)
- Confusion in classifying “T-shirt/top”, “Shirt”, and “Coat” classes