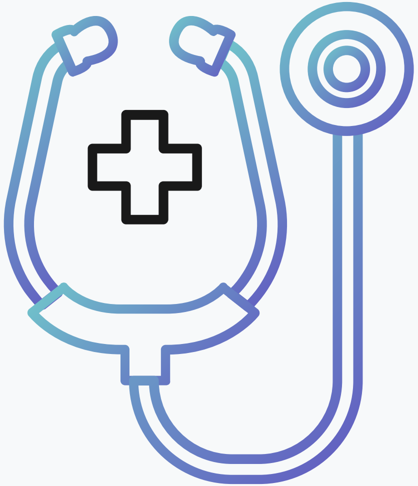
2XXX

WatPandas
Triage Classification



Kim Si Eun 1005370
Royden Yang 1005219
Samuel Chiang 1005142

Table of Contents

- → Problem
- → Datasets Used
- → Algorithms
- → Results
- → Conclusion

An Emergency Department Problem

Triage in the A&E plays the crucial role of identifying and prioritising patients with the most urgent needs to receive emergency service first.

Problem:
Lack of triage nurses on duty, can cause patients flow in A&E to bottleneck at triage stage.

Solution:

Automating the process of triage, with multi-model

Datasets Used

https://www.kaggle.com/datasets/maalona/hospital-triage-and-patient-history-data

Hospital Triage and Patient History dataset

https://www.kaggle.com/datasets/hossamahmedaly/patient-priority-classification

Patient Priority for Clustering

https://www.kaggle.com/datasets/erhmrai/ecg-image-data

ECG image dataset

Dataset 1 - Hospital Triage

- Originally 59k rows by 972 columns data
- Filter and kept relevant **chest pain** cases **(7k rows, 53 cols)**
- Age, gender, blood pressure, heart rate, other vitals, chest pain presence, chief complaints, etc
- Triage classified into 5 levels:
    - **Level 1:** immediate/life threatening
    - **Level 2:** emergency, could become life threatening
    - **Level 3:** urgent, not life threatening
    - **Level 4:** semi-urgent, not life threatening
    - **Level 5:** no resource required to stabilize patient

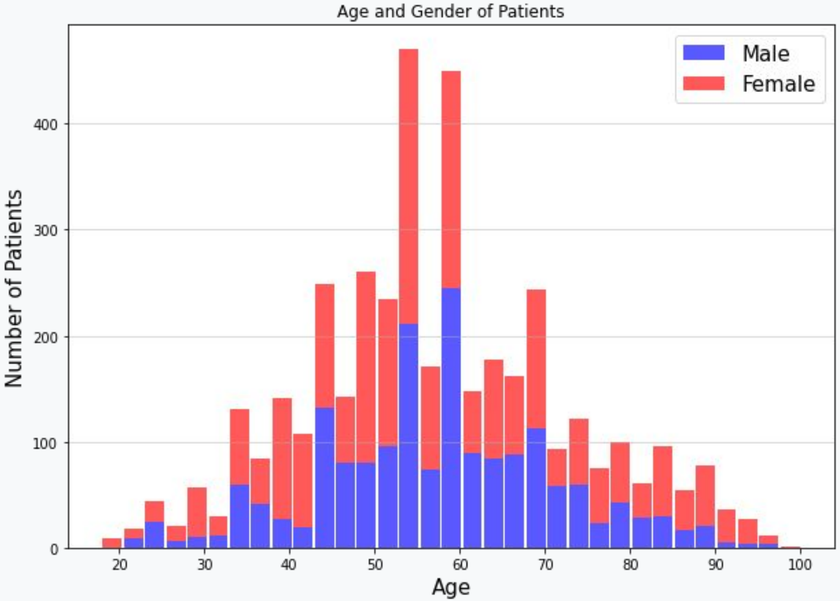
Dataset 1 - Hospital Triage

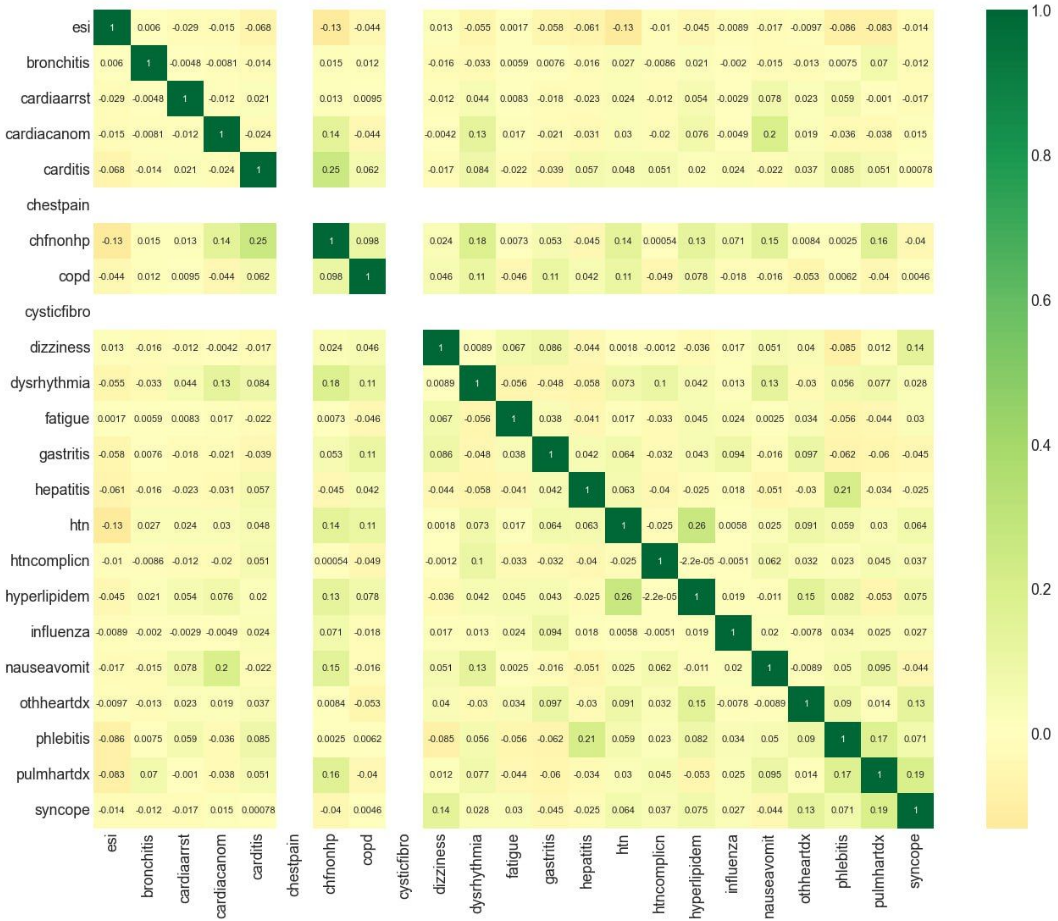
- Usage of correlograms to depict correlation metrics between multiple variables.

- **Symptoms:** Focus on immediately detectable symptoms, also relatively higher correlation values.

- Use of LogisticRegression and RandomForestClassifier classification algorithms for higher F1-scores.

- **F1-score**: Focus on false positives and false negatives, emphasis on the smaller ESI values (mainly 2 and 3 due to lack of data on 1) since identifying the emergency cases accurately are crucial.


Age and Gender of Patients
Male
Female
Number of Patients
Age
0
100
200
300
400
20
30
40
50
60
70
80
90
100


esi
bronchitis
cardiaarrst
cardiacanom
carditis
chestpain
chfnonhp
copd
cysticfibro
dizziness
dysrhythmia
fatigue
gastritis
hepatitis
htn
htncomplicn
hyperlipidem
influenza
nauseavomit
othheartdx
phlebitis
pulmhartdx
syncope
1.0
0.8
0.6
0.4
0.2
0.0

Symptoms Only

RandomForestClassifier (n_estimators = 100)

	precision	recall	f1-score	support
1.0	0.00	0.00	0.00	6
2.0	0.55	0.57	0.56	448
3.0	0.57	0.63	0.60	594
4.0	0.21	0.12	0.15	163
5.0	0.00	0.00	0.00	21
accuracy			0.53	1232
macro avg	0.27	0.26	0.26	1232
weighted avg	0.50	0.53	0.51	1232

Confusion Matrix:

```
[[  0   4   2   0   0]
 [  0 257 168  19   4]
 [  1 169 373  50   1]
 [  0  32 105  20   6]
 [  0   5   8   8   0]]
```

LogisticRegression (C=1.0)

	precision	recall	f1-score	support
1.0	0.00	0.00	0.00	6
2.0	0.56	0.44	0.50	448
3.0	0.54	0.80	0.64	594
4.0	0.00	0.00	0.00	163
5.0	0.00	0.00	0.00	21
accuracy			0.55	1232
macro avg	0.22	0.25	0.23	1232
weighted avg	0.46	0.55	0.49	1232

Confusion Matrix:

```
[[  0   2   4   0   0]
 [  0 199 249   0   0]
 [  0 120 474   0   0]
 [  0  30 133   0   0]
 [  0   4  17   0   0]]
```

Full Dataset

RandomForestClassifier (n_estimators = 100)

	precision	recall	f1-score	support
1.0	0.00	0.00	0.00	6
2.0	0.62	0.60	0.61	448
3.0	0.57	0.74	0.65	594
4.0	0.61	0.12	0.20	163
5.0	0.00	0.00	0.00	21
accuracy			0.59	1232
macro avg	0.36	0.29	0.29	1232
weighted avg	0.58	0.59	0.56	1232

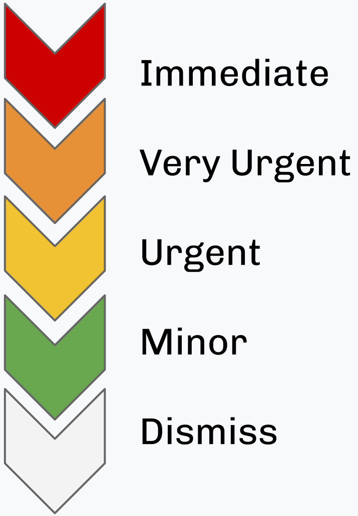
Dataset 1 - Hospital Triage

Values depicted do not look very high though?

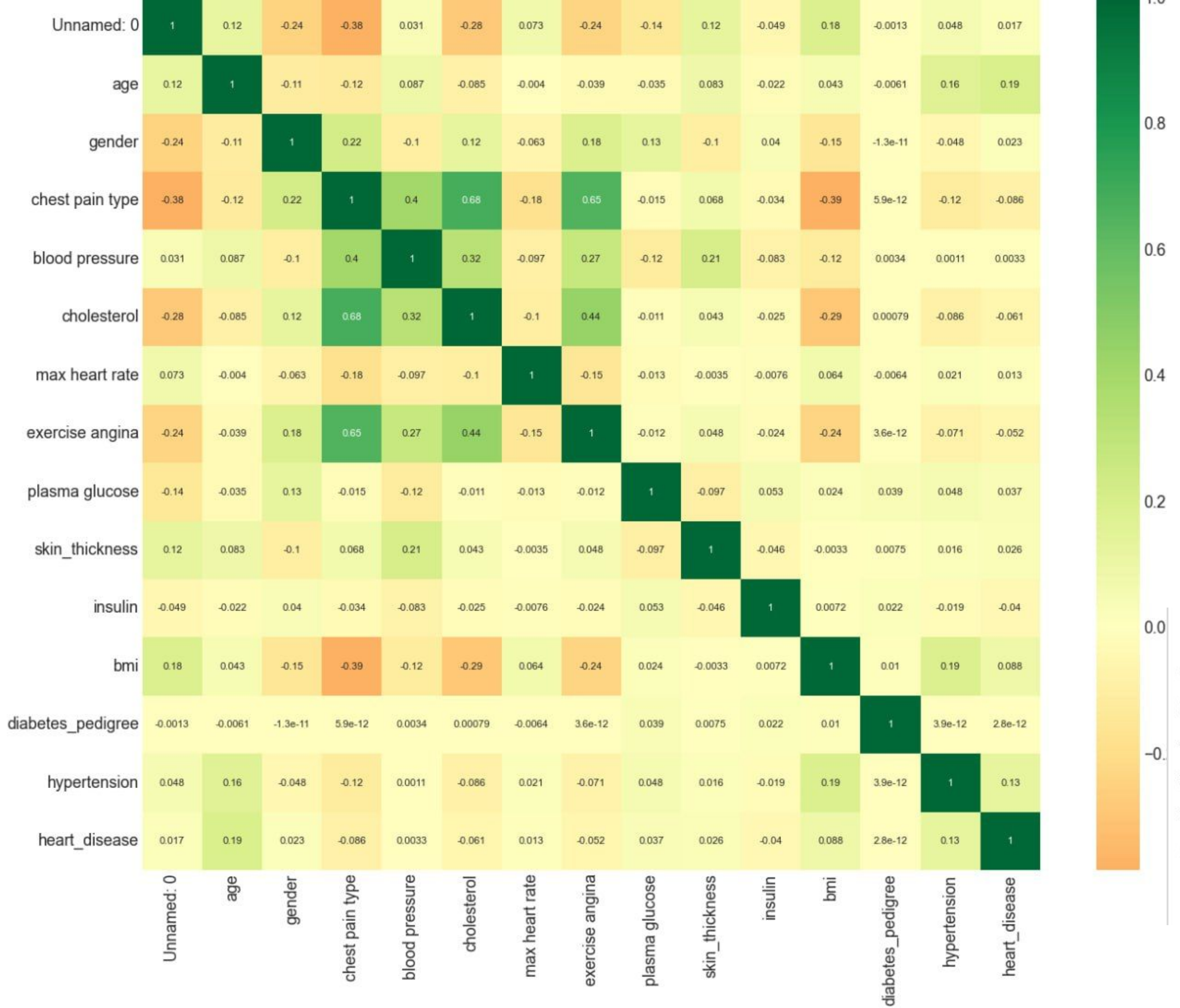
- Supposedly, the f1-scores obtained when classifying the full dataset as compared to symptoms only are higher, but the **values still aren't very high.**
- Possibly too many features being used in the classification: chief complaints (cc) was included in training, but what the patient said might not be very relevant to the actual diagnosis
- Yet, while focusing on the symptoms themselves, **correlation values and f1-scores are not significant enough.**
- **Conclusion:** This dataset alone is inconclusive in meeting our requirements for a solution.

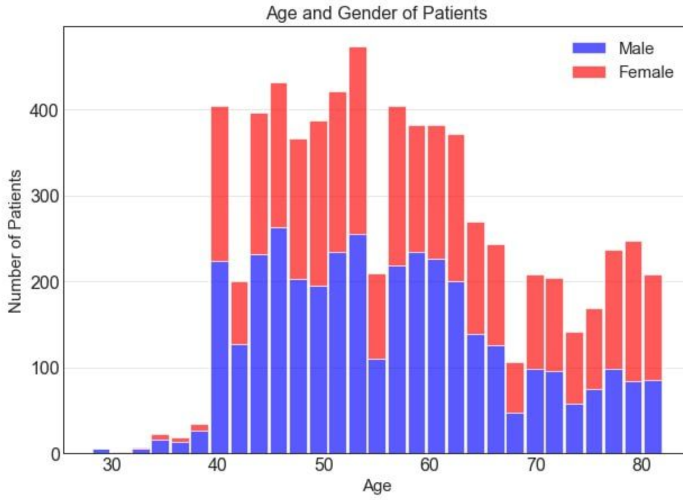
Dataset 2 - Patient Priority

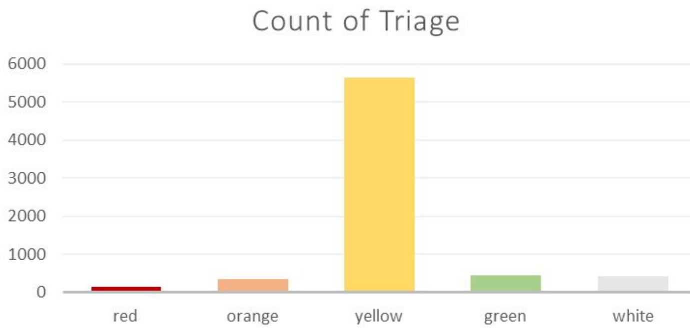
- **6962 rows by 17 columns** of data including blood pressure, heart rate, glucose levels, smoking status, etc
- Cleaned unnecessary rows such as “Residence type”
- 5 classifications of triage by color



age	gender	blood pressure	cholesterol	max heart rate	exercise angina	plasma glucose	skin thickness	insulin	bmi	diabetes pedigree	hyper tension	heart disease	smoking status	triage
60	1.0	121	160	184	0	153.48	78	113	37.3	0.467386	0	0	2	green
81	0.0	89	181	190	0	63.65	37	139	23.0	0.467386	0	0	0	yellow
50	1.0	82	153	155	0	90.42	50	91	16.2	0.467386	0	0	0	white
72	1.0	132	159	153	0	65.77	36	110	23.7	0.467386	0	0	3	yellow
70	0.0	93	183	180	0	87.52	88	123	39.2	0.467386	0	0	2	yellow


Unnamed: 0 | 1 | 0.12 | -0.24 | -0.38 | 0.031 | -0.28 | 0.073 | -0.24 | -0.14 | 0.12 | -0.049 | 0.18 | -0.0013 | 0.048 | 0.017
age | 0.12 | 1 | -0.11 | -0.12 | 0.087 | -0.085 | -0.004 | -0.039 | -0.035 | 0.083 | -0.022 | 0.043 | -0.0061 | 0.16 | 0.19
gender | -0.24 | -0.11 | 1 | 0.22 | -0.1 | 0.12 | -0.063 | 0.18 | 0.13 | -0.1 | 0.04 | -0.15 | -1.3e-11 | -0.048 | 0.023
chest pain type | -0.38 | -0.12 | 0.22 | 1 | 0.4 | 0.68 | -0.18 | 0.65 | -0.015 | 0.068 | -0.034 | -0.39 | 5.9e-12 | -0.12 | -0.086
blood pressure | 0.031 | 0.087 | -0.1 | 0.4 | 1 | 0.32 | -0.097 | 0.27 | -0.12 | 0.21 | -0.083 | -0.12 | 0.0034 | 0.0011 | 0.0033
cholesterol | -0.28 | -0.085 | 0.12 | 0.68 | 0.32 | 1 | -0.1 | 0.44 | -0.011 | 0.043 | -0.025 | -0.29 | 0.00079 | -0.086 | -0.061
max heart rate | 0.073 | -0.004 | -0.063 | -0.18 | -0.097 | -0.1 | 1 | -0.15 | -0.013 | -0.0035 | -0.0076 | 0.064 | -0.0064 | 0.021 | 0.013
exercise angina | -0.24 | -0.039 | 0.18 | 0.65 | 0.27 | 0.44 | -0.15 | 1 | -0.012 | 0.048 | -0.024 | -0.24 | 3.6e-12 | -0.071 | -0.052
plasma glucose | -0.14 | -0.035 | 0.13 | -0.015 | -0.12 | -0.011 | -0.013 | -0.012 | 1 | -0.097 | 0.053 | 0.024 | 0.039 | 0.048 | 0.037
skin_thickness | 0.12 | 0.083 | -0.1 | 0.068 | 0.21 | 0.043 | -0.0035 | 0.048 | -0.097 | 1 | -0.046 | -0.0033 | 0.0075 | 0.016 | 0.026
insulin | -0.049 | -0.022 | 0.04 | -0.034 | -0.083 | -0.025 | -0.0076 | -0.024 | 0.053 | -0.046 | 1 | 0.0072 | 0.022 | -0.019 | -0.04
bmi | 0.18 | 0.043 | -0.15 | -0.39 | -0.12 | -0.29 | 0.064 | -0.24 | 0.024 | -0.0033 | 0.0072 | 1 | 0.01 | 0.19 | 0.088
diabetes_pedigree | -0.0013 | -0.0061 | -1.3e-11 | 5.9e-12 | 0.0034 | 0.00079 | -0.0064 | 3.6e-12 | 0.039 | 0.0075 | 0.022 | 0.01 | 1 | 3.9e-12 | 2.8e-12
hypertension | 0.048 | 0.16 | -0.048 | -0.12 | 0.0011 | -0.086 | 0.021 | -0.071 | 0.048 | 0.016 | -0.019 | 0.19 | 3.9e-12 | 1 | 0.13
heart_disease | 0.017 | 0.19 | 0.023 | -0.086 | 0.0033 | -0.061 | 0.013 | -0.052 | 0.037 | 0.026 | -0.04 | 0.088 | 2.8e-12 | 0.13 | 1
Unnamed: 0
age
gender
chest pain type
blood pressure
cholesterol
max heart rate
exercise angina
plasma glucose
skin_thickness
insulin
bmi
diabetes_pedigree
hypertension
heart_disease
0.8
0.6
0.4
0.2
0.0


Age and Gender of Patients
Male
Female
Number of Patients
0
100
200
300
400
30
40
50
60
70
80
Age


Count of Triage
6000
5000
4000
3000
2000
1000
0
red
orange
yellow
green
white

Dataset 2 - Patient Priority

- Experimented with different classification algorithms (Decision Tree, Logistic Regression, K-NN, Gaussian Naive Bayes, SVM, Ada Boost)
- Focused on **F1-score** since false positives and false negatives are more important, and we have uneven distribution of classes.

Dataset 2 Experimentation

Decision Tree (depth = 10)

	precision	recall	f1-score	support
red	0.76	0.79	0.78	33
orange	0.81	0.85	0.83	93
yellow	0.99	0.99	0.99	1695
green	1.00	0.99	1.00	137
white	0.94	0.97	0.95	131
accuracy			0.98	2089
macro avg	0.90	0.92	0.91	2089
weighted avg	0.98	0.98	0.98	2089

```
Confusion Matrix:
[[  26    2    3    0    2]
 [   4   79   10    0    0]
 [   2   16 1671    0    6]
 [   0    0    1  136    0]
 [   2    1    1    0  127]]
```

Ada Boost (estimators = 1500, learning rate=0.05

	precision	recall	f1-score	support
red	0.81	0.64	0.71	33
orange	0.76	0.58	0.66	93
yellow	0.97	0.99	0.98	1695
green	1.00	0.99	1.00	137
white	0.91	0.92	0.91	131
accuracy			0.96	2089
macro avg	0.89	0.82	0.85	2089
weighted avg	0.96	0.96	0.96	2089

```
Confusion Matrix:
[[  21    1    8    0    3]
 [   4   54   34    0    1]
 [   0   13 1674    0    8]
 [   0    0    1  136    0]
 [   1    3    7    0  120]]
```

Really bad to classify red levels as white and vice versa!

Merge Red&Orange into Immediate

Decision Tree (depth = 10)

	precision	recall	f1-score	support
immediate	0.81	0.93	0.87	126
yellow	0.99	0.98	0.99	1695
green	1.00	0.99	1.00	137
white	0.96	0.96	0.96	131
accuracy			0.98	2089
macro avg	0.94	0.97	0.95	2089
weighted avg	0.98	0.98	0.98	2089

```
Confusion Matrix:
 [[ 117    7    0    2]
 [   24 1668    0    3]
 [    0    1  136    0]
 [    3    2    0  126]]
```

Ada Boost (n_estimators=1500, learning_rate=0.06)

	precision	recall	f1-score	support
immediate	0.76	0.81	0.78	126
yellow	1.00	0.98	0.99	1695
green	1.00	0.99	1.00	137
white	0.82	0.95	0.88	131
accuracy			0.97	2089
macro avg	0.89	0.93	0.91	2089
weighted avg	0.97	0.97	0.97	2089

```
Confusion Matrix:
 [[ 102    3    0   21]
 [   30 1658    0    7]
 [    0    1  136    0]
 [    3    4    0  124]]
```

Higher f1 scores, still contain some misclassifications

Combine Datasets 1&2

- Common columns of both datasets: **age, gender, glucose level, heart rate, blood pressure**
- Experimented with classification algorithms using combined dataset

	esi	age	gender	glucose	heart rate	blood pressure
0	3	66.0	1	132.00	66.0	136.0
1	2	80.0	1	96.00	70.0	112.0
2	3	80.0	1	95.00	95.0	126.0
3	2	80.0	1	94.00	64.0	132.0
4	3	80.0	1	96.00	91.0	133.0
...	...	...	...	...	...	...
11063	3	80.0	0	83.75	166.0	111.0
11064	3	81.0	0	125.20	160.0	123.0
11065	3	81.0	0	82.99	141.0	127.0
11066	4	51.0	1	166.29	162.0	123.0
11067	3	44.0	0	85.28	172.0	125.0

Combine Datasets 1&2

- Same classification algorithms as previously, but showed very poor results;
  f1 scores < 0.5
- Attempted simple neural network model with 2 layers

```
# keras model
model = Sequential()          # a model consisting of successive layers
# input layer
model.add(Dense(n_neurons, input_dim=input_dim, activation='sigmoid'))
# output layer, with 5 neuron
model.add(Dense(5, activation='sigmoid'))
# compile the model
model.compile(loss='mean_squared_error', optimizer='sgd', metrics=['accuracy'])
```

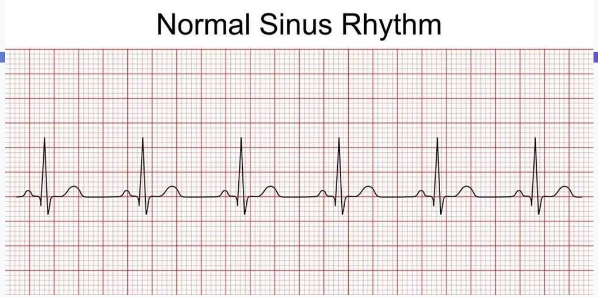
```
results = model.evaluate(final_X_test, final_y_test)
print("test loss, test acc:", results)
```

✓ 0.4s

```
104/104 [==============================] - 0s 1ms/step - loss: 0.0869 - accuracy: 0.6949
test loss, test acc: [0.08686400204896927, 0.6948795318603516]
```

Suspicion: **5 features are not enough** to classify triage accurately, even if combining datasets gave more datapoints.

ECG Images for Triage



- Electrocardiogram readings are often taken by nurses during triage, especially in presence of chest pain symptoms.

- Providing vital information about cardiac rhythm, presence of arrhythmias, myocardial ischemia/infarction, and other abnormalities.

- Important in discovering possibility of heart attacks

ECG Image Dataset

Train Data : 99,190 images
75k of which were Normal beat

Test Data : 24,773 images
18k of which were Normal beat

6 classifications

N: Normal beat

S: Supraventricular premature beat

V: Premature ventricular contraction

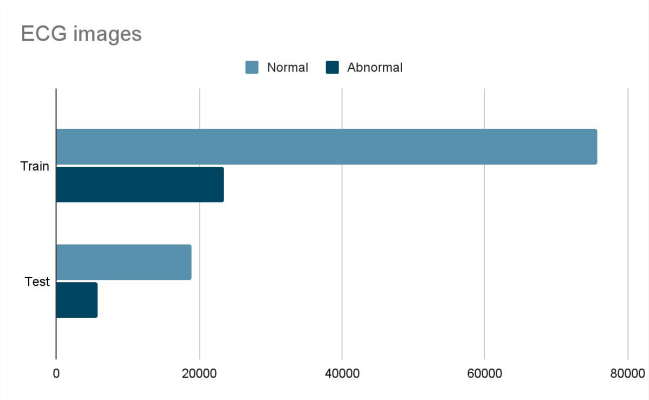
F: Fusion of ventricular and normal beat

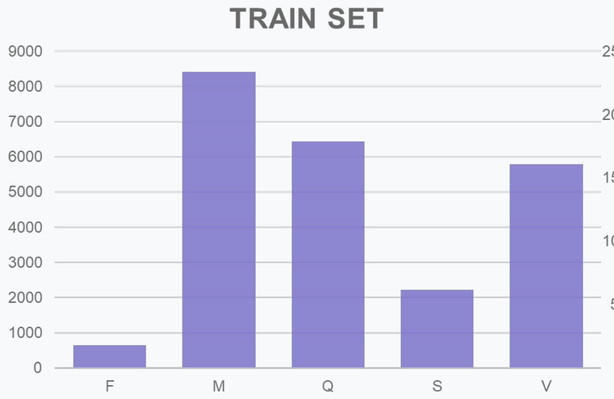
Q: Unclassifiable beat

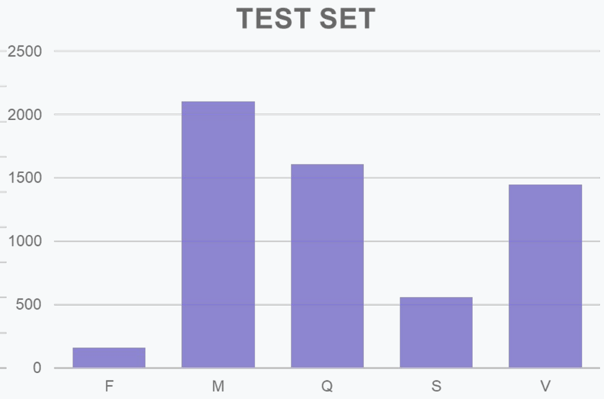
M: myocardial infarction

ECG Image Dataset

Convolutional NN with PyTorch







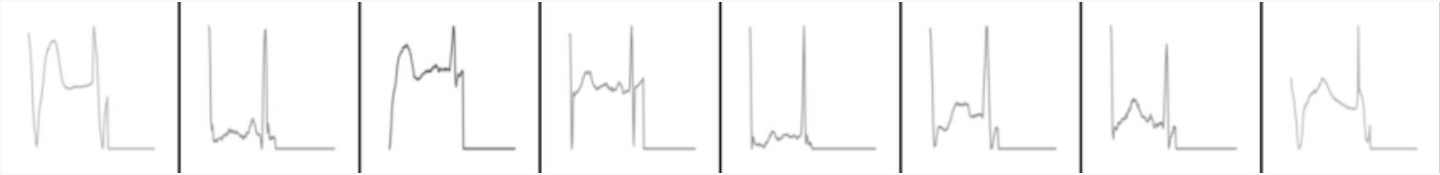
Normal : Abnormal
4:1

Preprocessing Images

In the original dataset, images from different classes had different colors, hence training CNN as such would lead to 100% accuracy

Scaling down image sizes for faster training

```
train_dataset = ImageFolder(train_folder,transform = transforms.Compose([
    transforms.Resize((150,150)),
    transforms.Grayscale(num_output_channels=1),
    transforms.ToTensor()
]))
```



ECG_Classifier Model

Started with a small sequential block

```
nn.Conv2d(1, 16, kernel_size = 3, padding = 1),
nn.ReLU(),
nn.Conv2d(16,32, kernel_size = 3, stride = 1, padding = 1),
nn.ReLU(),
nn.MaxPool2d(2,2),
```

Followed by flattening and passing through linear layer

```
nn.Flatten(),
nn.ReLU(),
nn.Linear(1024,6)
```

However, training extremely long and encountered GPU memory issues.

ECG_Classifier Model

```
nn.Conv2d(1, 32, kernel_size = 3, padding = 1),
nn.BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_runni
nn.ReLU(),
nn.Conv2d(32,64, kernel_size = 3, stride = 1, padding = 1),
nn.BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_runni
nn.ReLU(),
nn.MaxPool2d(2,2),
```

New Block
Introduced BatchNorm2d layers which accelerate training

Taking outputs from a layer and normalize them before passing them on as inputs to the next layer.

Introduced Dropout layers to make network simpler as training progresses

```
nn.Dropout(p=0.5, inplace=False),
nn.Linear(1024,6)
```

Stacking 5 blocks

```
nn.Conv2d(1, 32, kernel_size = 3, padding = 1),
nn.BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True),
nn.ReLU(),
nn.Conv2d(32,64, kernel_size = 3, stride = 1, padding = 1),
nn.BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True),
nn.ReLU(),
nn.MaxPool2d(2,2),
```

⋮

```
nn.Conv2d(512, 512, kernel_size = 3, stride = 1, padding = 1),
nn.BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True),
nn.ReLU(),
nn.Conv2d(512,512, kernel_size = 3, stride = 1, padding = 1),
nn.BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True),
nn.ReLU(),
nn.MaxPool2d(2,2),


nn.Flatten(),
nn.Linear(8192,4096),
nn.ReLU(),
nn.Dropout(p=0.5, inplace=False),
nn.Linear(4096, 1024),
nn.ReLU(),
nn.Dropout(p=0.5, inplace=False),
nn.Linear(1024,6)
```

ECG_Classifier Model

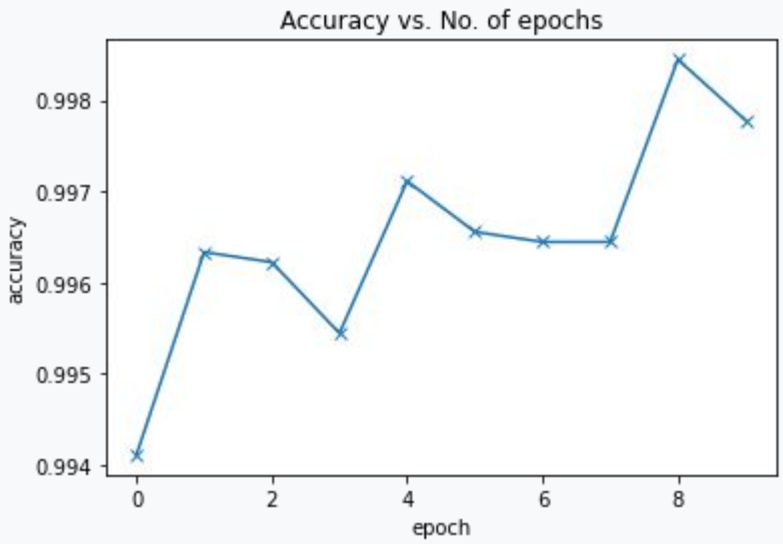
```
train_data,val_data = random_split(train_dataset,[train_size,val_size])
print(f"Train size : {len(train_data)}")
print(f"Validation Size : {len(val_data)}")


#load the train and validation into batches.
train_dataloader = DataLoader(train_data, batch_size, shuffle = True, num_workers = 4)
val_dataloader = DataLoader(val_data, batch_size, shuffle = True, num_workers = 4)
```

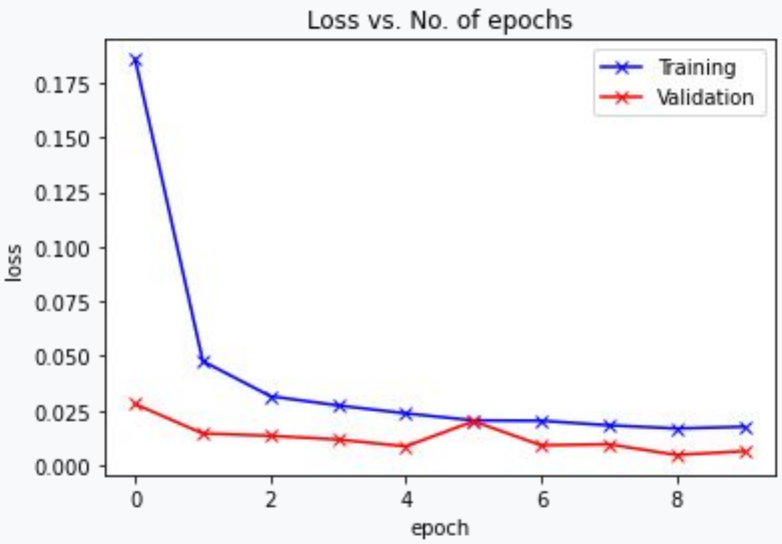
Optimizer : Adam optimizer
Loss Func : Cross Entropy
Learning Rate : 0.001
Epochs : 10
Batch Size : 8
Train Validation Random Split : 90% Train, 10% Validation

model.train() on Train batch, model.eval() on Validation batch each epoch

```
Epoch [7], train_loss: 0.0182, val_loss: 0.0095, val_acc: 0.9964
Epoch 9/10, Loss: 0.0000:  63%|██████     | 7126/11275 [08:28<05:10, 13.37it/s]
Epoch [8], train_loss: 0.0168, val_loss: 0.0046, val_acc: 0.9984
Epoch 10/10, Loss: 0.0000:  92%|█████████ | 10334/11275 [12:00<01:14, 12.68it/s]
Epoch [9], train_loss: 0.0175, val_loss: 0.0064, val_acc: 0.9978
```

ECG_Classifier Model Training





Validation Accuracy: 99.78%

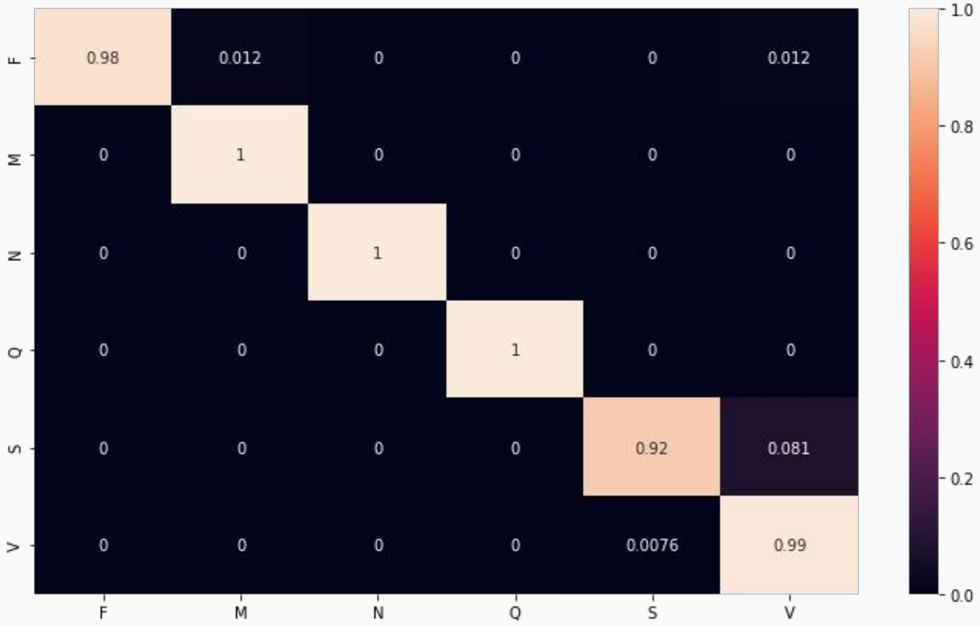
Testing on 24k img

Test Accuracy: 99.75805476027259%

Classification Report

	precision	recall	f1-score	support
F	1.00	0.98	0.99	161
M	1.00	1.00	1.00	2101
N	1.00	1.00	1.00	18926
Q	1.00	1.00	1.00	1608
S	0.98	0.92	0.95	556
V	0.97	0.99	0.98	1447
accuracy			1.00	24799
macro avg	0.99	0.98	0.99	24799
weighted avg	1.00	1.00	1.00	24799

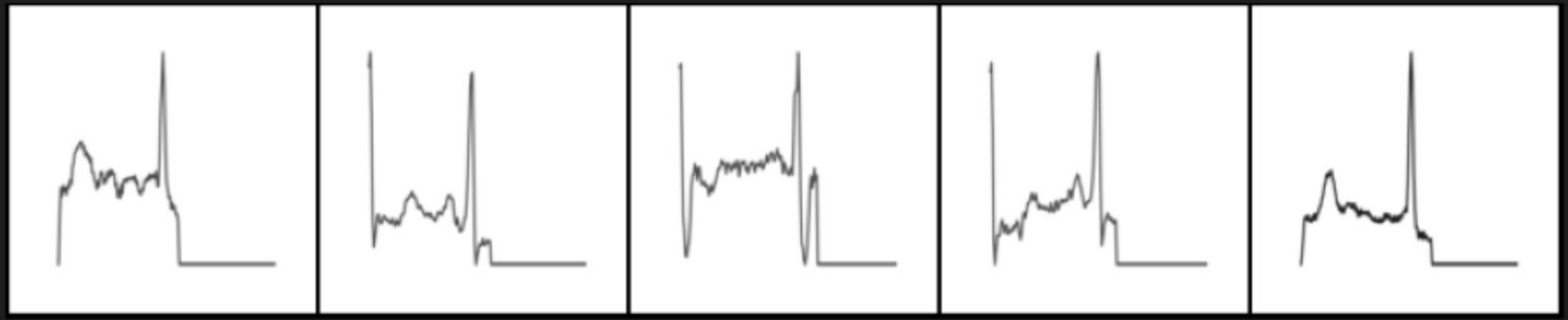
Confusion Matrix



Error cases

Total 60 images failed in Test Set, displaying first 5 cases

```
ECG Image 1 Actual class :  S , but predicted as :  V
ECG Image 2 Actual class :  S , but predicted as :  V
ECG Image 3 Actual class :  S , but predicted as :  V
ECG Image 4 Actual class :  S , but predicted as :  V
ECG Image 5 Actual class :  F , but predicted as :  V
```



Error cases

Classes with misclassifications were F, S and V.
All Normal beat ECG were predicted correctly.

```
The incorrect classification counts are :
 F : 4 , M : 0 , N : 0 , Q : 0 , S : 45 , V : 11
```

```
Abnormal classified incorrectly : 60
Normal classified incorrectly : 0
```

Looking at the confusion matrix,
There were **no False Negatives** where abnormal ECG were classified as normal.

Conclusion

Dataset 1.... This dataset as a standalone does not provide conclusive enough results for us to adopt as our primary solution.

Dataset 2...Lower levels of yellow, green, and white, performed very accurately, but higher levels of red and orange performed relatively less accurate.

For ECG images, the CNN model could perform sufficiently well and F1 score was high with no false negatives.
Can correctly distinguish and output Normal vs Abnormal cases.

Future Improvements

To combine the datasets of recordings, images, etc.

Need datasets with proper labels and feature inputs that can be encapsulated by the model.

Dataset that uses both the usual triage features inputs as well as patient’s scan readings, classifying model that can combine both to accurately triage further.

10 Q