# Project Report : ECG Image-based Heartbeat classification for Arrhythmia Detection Using IBM Watson Studio

By: Ishika Naik

## 1 INTRODUCTION

### 1.1 Overview

Cardiovascular disease (CVD) is one of the main causes of death in the world today. Analyzing the Electrocardiogram (ECG), a medical monitoring device that records heart activity, is the current method of disease detection. Unfortunately, finding professionals to examine a big volume of ECG data takes up far too much medical time and money. As a result, machine learning-based methods for recognising ECG characteristics have steadily gained traction. These traditional methods, on the other hand, have several downsides, such as the need for manual feature recognition, complex models, and a long learning curve.

The five micro-classes of heartbeat types in the MIT-BIH Arrhythmia database are classified using a robust and efficient 7-layer deep one-dimensional convolutional neural network. The results reveal that the model used for this project has high accuracy, sensitivity, resilience, and anti-noise capabilities. Its precise classification effectively conserves medical resources, which can benefit clinical practise.

### 1.2 Purpose

Cardiovascular disease is a frequent condition that poses a major threat to human health, particularly in the middle-aged and elderly. It has a high prevalence, a high level of disability, and a high death rate. The globe is currently dealing with a population that is ageing. Cardiovascular disease is becoming a big public health issue as it continues to worsen. ECG analysis is a good approach to check your heart's health. As a result, cardiovascular disorders necessitate the detection and classification of ECG signals. Not just for early identification and treatment, but also for early prevention. The classification of linked ECG signals is extremely important to investigate.

The website interface created to facilitate the classification of signal images can be used to quickly and efficiently to identify the presence of Arrhythmia in a patient and also classify the specific type of Arrhythmia the patient is suffering from out of the following 6 categories:

- Left Bundle Branch Block
- Normal
- Premature Atrial Contraction
- Premature Ventricular Contractions
- Right Bundle Branch Block
- Ventricular Fibrillation

By uploading an image of the ECG signal, medical facilities around the world can generate predictions instantly, which can save money as well as time, something that may be crucial for a patient that requires immediate surgery. It also provides an efficient solution to remote diagnosis, especially in a scenario where a health professional isn't available.

# 2 LITERATURE SURVEY

## 2.1 Existing problem

There are several existing methods to detect the presence of Arrhythmia. An electrocardiogram (ECG or EKG). During an ECG, sensors (electrodes) that can detect the electrical activity of the heart are attached to the chest and sometimes to the arms or legs. An ECG measures the timing and duration of each electrical phase in the heartbeat. It is up to the health professional to determine the presence through previous experience and medical knowledge.

## 2.2 Proposed solution

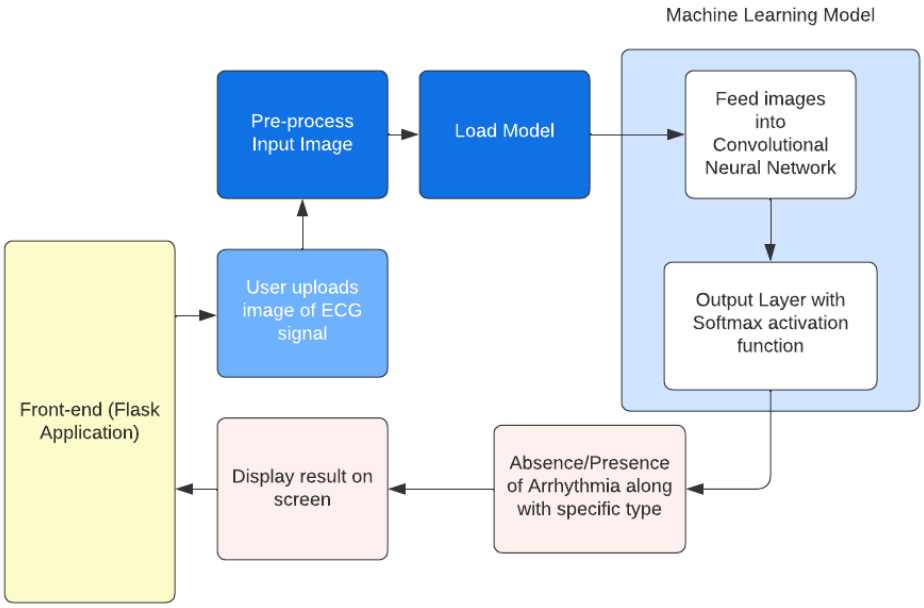
The proposed solution represents portions of these signals as images, the shape of which can be used to train a deep convolutional neural network. A Convolutional Neural Network is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other. A Convolutional neural network is able to successfully capture the Spatial and Temporal dependencies in an image through the application of relevant filters. The architecture performs a better fitting to the image dataset due to the reduction in the number of parameters involved and reusability of weights. In other words, the network can be trained to understand the sophistication of the image better.

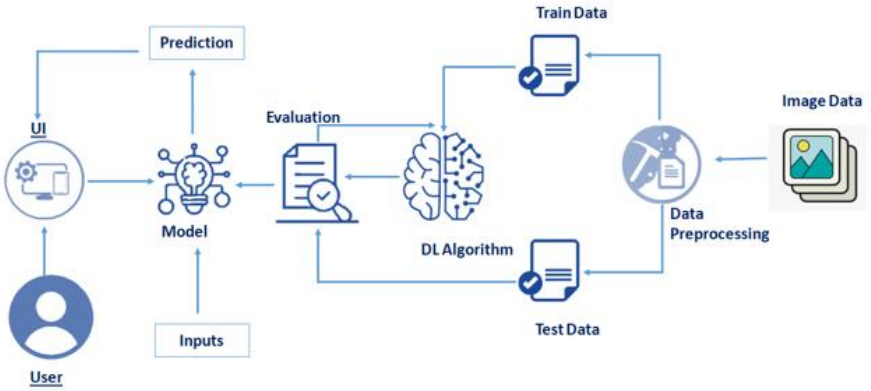
Hence, we can make use of these to train a CNN which can classify images into the seven possible classes; the six categories of Arrhythmia and the normal class. The images are pre-processed and reshaped to a 64 x 64 dimension. This can be fed into a sequential model, where we use Max-Pooling layers and Convolutional layers to extract relevant features. A ‘ReLu’ activation function is used. Finally, we flatten the 2-D output of the Convolutional layer and use it to determine which of the 7 classes score the highest by using a ‘Softmax’ activation function.

The class that gets outputted is the predicted class, which we can use to infer the presence/absence and the type of arrhythmia.

# 3 THEORITICAL ANALYSIS

## 3.1 Block diagram




## 3.2 Hardware / Software designing

**Hardware requirements**: Operating System: Windows/Linux/Mac OS, Memory (RAM): Minimum 1 GB; Recommended 4 GB or above, Ethernet connection (LAN) OR a wireless adapter (Wi-Fi)

**Software requirements**:

1. IBM Watson Studio
2. Flask Framework
3. Tensorflow
4. Keras
5. HTML
6. CSS
7. Python

# 4 EXPERIMENTAL INVESTIGATIONS

The "training" of the neural network refers to the process of altering the weights' values. To begin, the CNN starts with random weights. During CNN training, the neural network is fed a huge dataset of images that have been labeled with their respective class labels. The CNN network processes each image by assigning random values to its pixels and then comparing them to the input image's class label. If the output does not match the class label (which usually happens at the start of the training phase), it makes a tiny change to the weights of its CNN neurons to ensure that the output matches the class label picture appropriately.

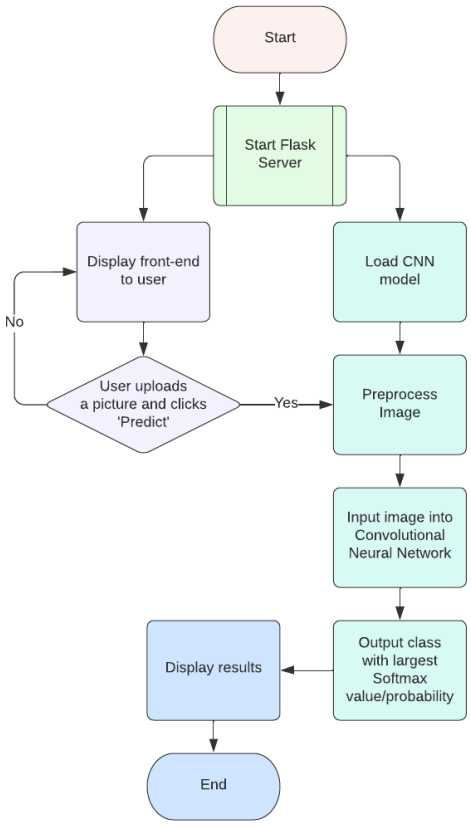
During model training it was observed that the model attained an accuracy of about 92% after being trained over a span of 10 epochs and hence, the model was able to distinguish between all 7 output classes with a relatively high accuracy.

```
In [21]: # Fit the model
         model.fit_generator(generator=x_train,steps_per_epoch = len(x_train),
                             epochs=10, validation_data=x_test,validation_steps = len(x_test))
```

```
/tmp/wsuser/ipykernel_2832/1916141677.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future ve
rsion. Please use `Model.fit`, which supports generators.
  model.fit_generator(generator=x_train,steps_per_epoch = len(x_train),
```

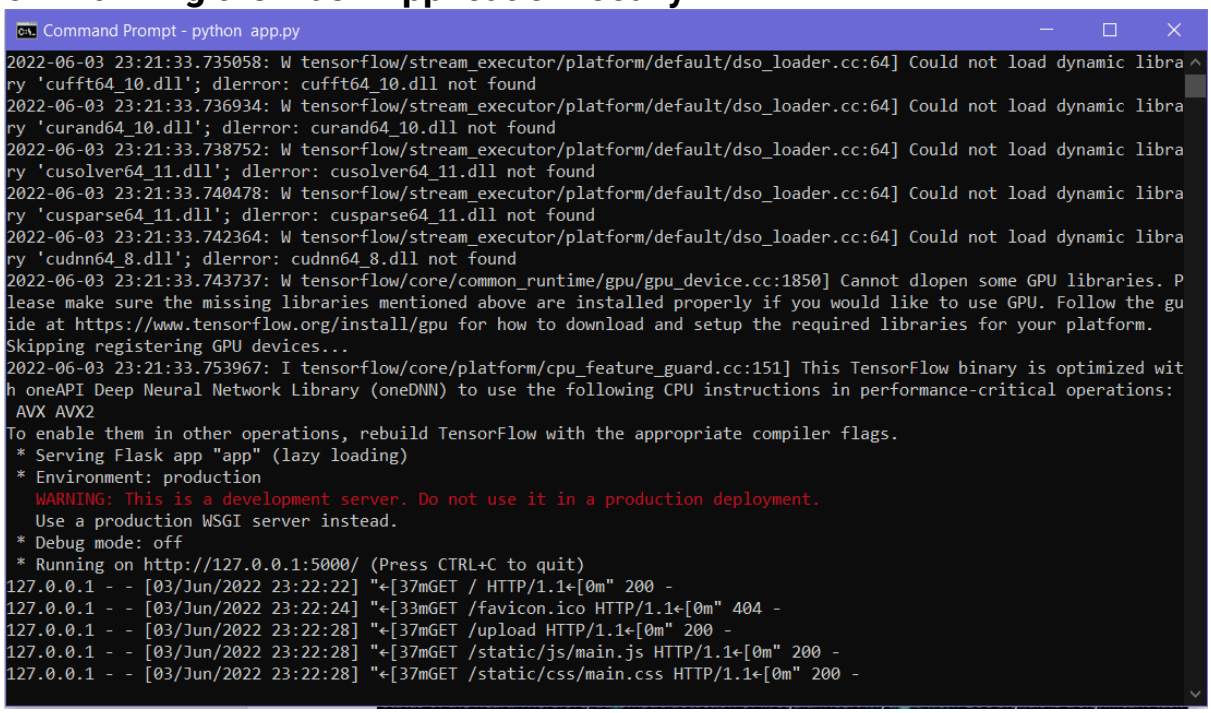
```
Epoch 1/10
480/480 [==============================] - 97s 201ms/step - loss: 0.7450 - accuracy: 0.7491 - val_loss: 0.4413 - val_accuracy:
0.8504
Epoch 2/10
480/480 [==============================] - 96s 201ms/step - loss: 0.2724 - accuracy: 0.9186 - val_loss: 0.4429 - val_accuracy:
0.8809
Epoch 3/10
480/480 [==============================] - 97s 201ms/step - loss: 0.2204 - accuracy: 0.9347 - val_loss: 0.2756 - val_accuracy:
0.9207
Epoch 4/10
480/480 [==============================] - 97s 201ms/step - loss: 0.1896 - accuracy: 0.9435 - val_loss: 0.2367 - val_accuracy:
0.9314
Epoch 5/10
480/480 [==============================] - 96s 201ms/step - loss: 0.1637 - accuracy: 0.9497 - val_loss: 0.2607 - val_accuracy:
0.9174
Epoch 6/10
480/480 [==============================] - 96s 200ms/step - loss: 0.1526 - accuracy: 0.9544 - val_loss: 0.2423 - val_accuracy:
0.9263
Epoch 7/10
480/480 [==============================] - 96s 199ms/step - loss: 0.1366 - accuracy: 0.9567 - val_loss: 0.2960 - val_accuracy:
0.9169
Epoch 8/10
480/480 [==============================] - 97s 202ms/step - loss: 0.1232 - accuracy: 0.9608 - val_loss: 0.3316 - val_accuracy:
0.9078
Epoch 9/10
480/480 [==============================] - 97s 201ms/step - loss: 0.1255 - accuracy: 0.9626 - val_loss: 0.2454 - val_accuracy:
0.9297
Epoch 10/10
480/480 [==============================] - 95s 199ms/step - loss: 0.1174 - accuracy: 0.9638 - val_loss: 0.3265 - val_accuracy:
0.9182
```

# 5 FLOWCHART

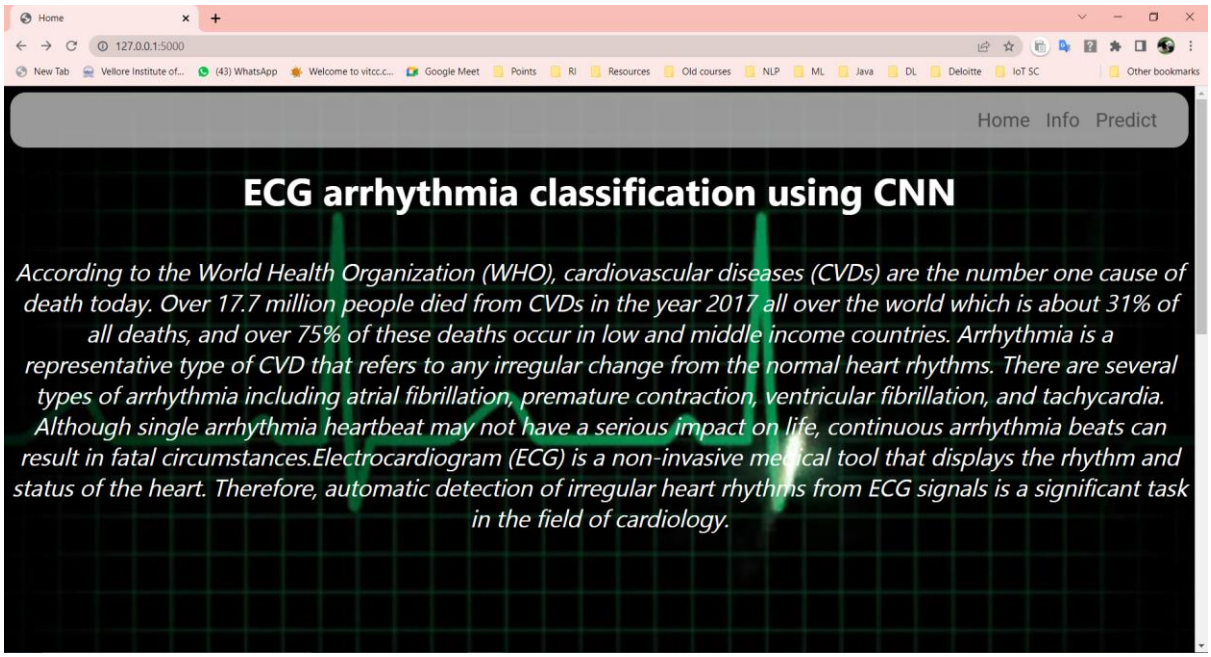



# 6 RESULT

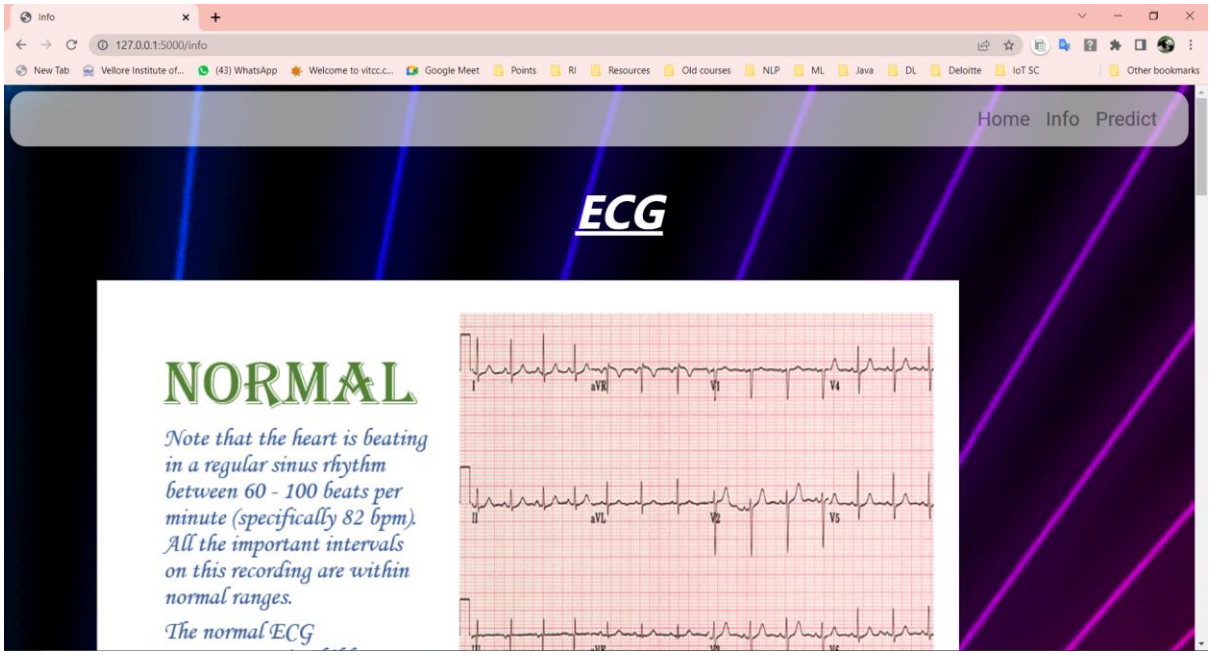
## 6.1 Running the Flask Application locally:



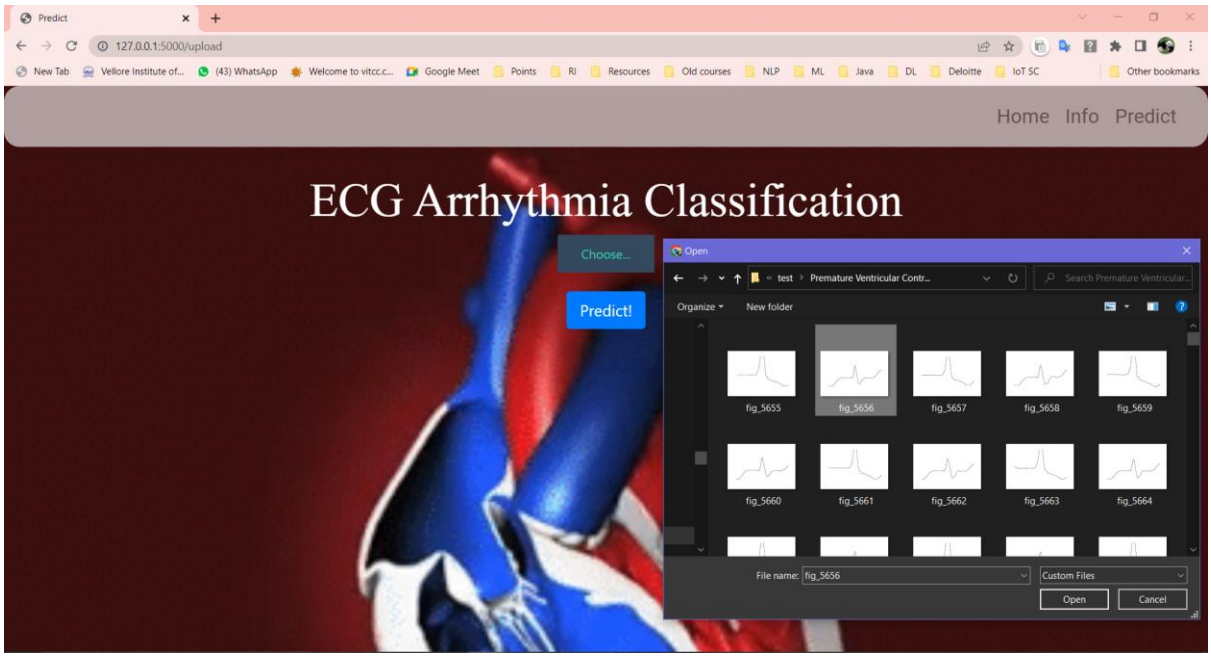
## The Homepage



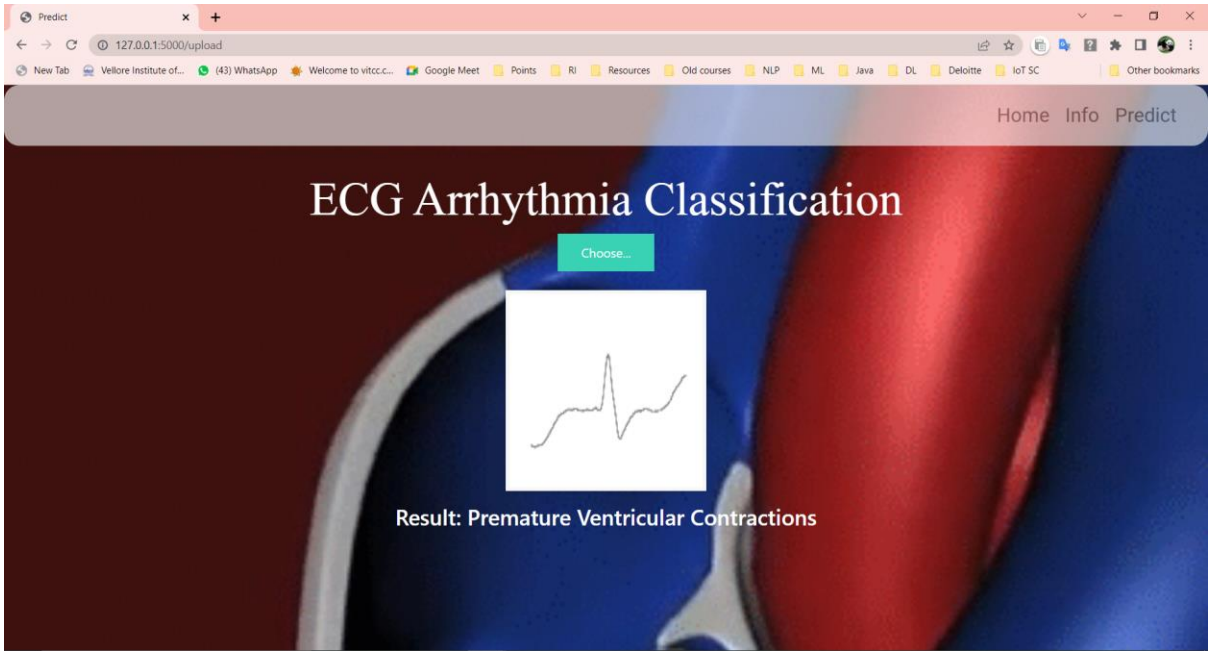

## Information Page



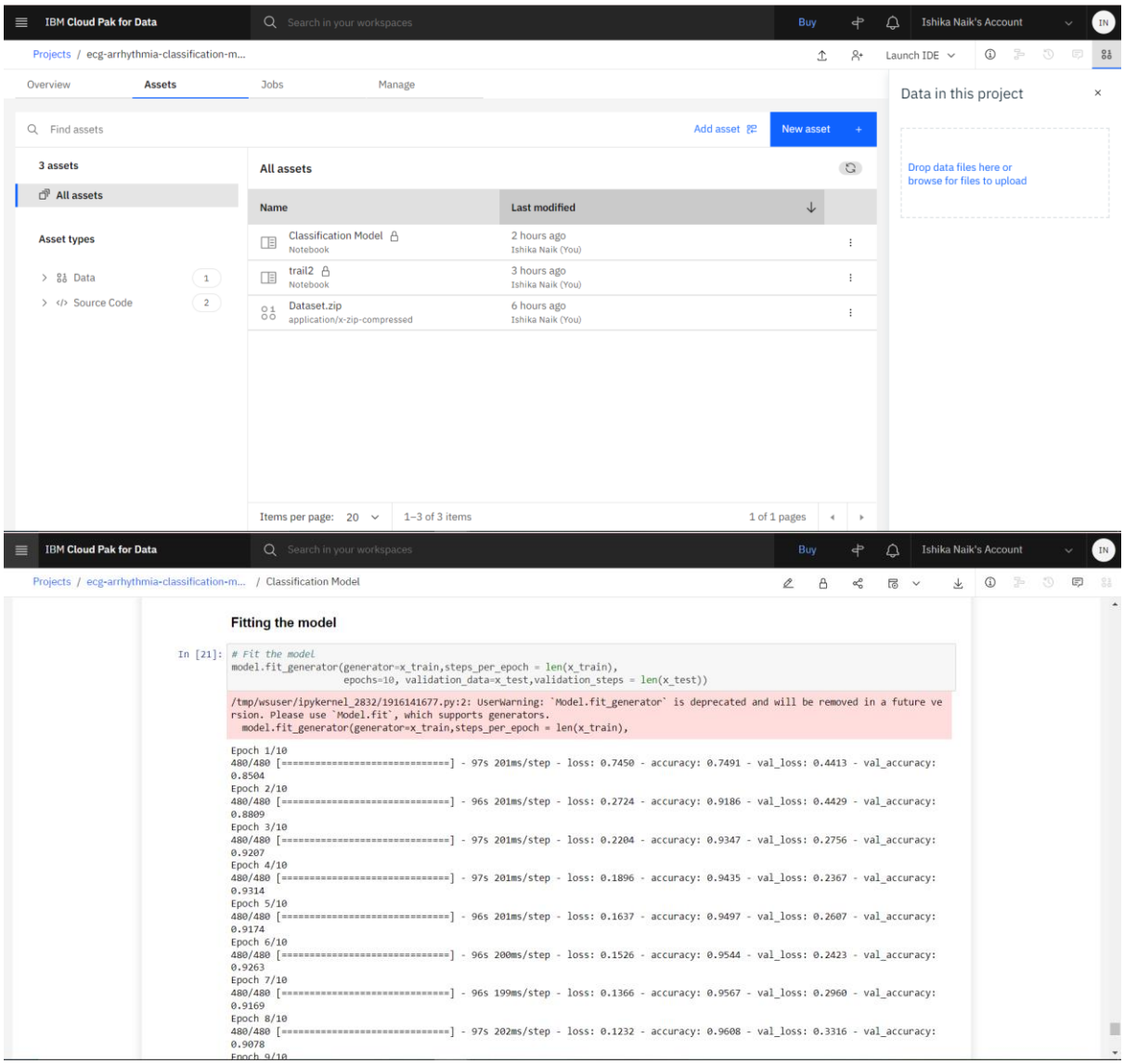
## Prediction Page




## Prediction Generation and displaying the result



## 6.2 IBM Watson Studio




# 7 ADVANTAGES & DISADVANTAGES

Advantages include:

1. The main advantage of CNN compared to its predecessors is that it automatically detects the important features without any human supervision.
2. CNN is also computationally efficient. It uses special convolution and pooling operations and performs parameter sharing. This enables CNN models to run on any device, making them universally attractive.
3. The interface is simple to use and can be used from any location in any circumstance as long as the user has access to a laptop.
4. Can provide a cheap alternative to in-person consultations and allow for diagnosis in situations where healthcare professionals are not available.

5. Use of GPUs can execute massively parallel computations that are scalable for vast amounts of data. Furthermore, when the volume of data is large, it produces higher performance results.

Disadvantages include:

1. The application cannot guarantee 100% accuracy and hence an incorrect diagnosis could be fatal for the patient.
2. Application requires the availability of an internet connection, a laptop, and the ECG. Without these, the prediction cannot be generated and hence this is not useful for remote areas where the network signal is weak and there are frequent power cuts.
3. One of the main barriers for the wide adoption of deep learning methods in clinical practice is usually caused by the variability in the data itself (e.g., contrast, resolution, signal-to-noise) and in this case, the ECG signals.
4. Deep learning models usually suffer from poor generalization when used with input data that comes from different machines (different vendor, model, etc.), with different acquisition parametrization or any underlying component that can cause the data distribution to shift. Hence, a difference in the ECG signal parameters while obtaining a pictorial representation can cause unintended inaccuracies.
5. These over-parametrized models have a high tendency to rely on superficial statistical patterns of the data, and are not immune to the distributional shift caused by external factors (different scanner, different acquisition protocol, etc.).

## 8 APPLICATIONS

This application can be used in a wide variety of scenarios. It can be used to assist the decision-making process of a doctor. If the predicted class matches what the doctor has diagnosed, the doctor may confidently administer the required treatment for the condition. However, if the model shows a difference between what the doctor has diagnosed and what it has predicted, it would be wiser to get a second opinion. This way we can ensure a highly accurate diagnosis where we rely on both humans and computers. In another scenario where a doctor is not available but a nurse is, the nurse can take the ECG readings and use the application to generate a prediction hence removing the need for the presence of a doctor. Since there is a general shortage of doctors, this can be useful for diagnosis in rural areas etc.

## 9 CONCLUSION

Based on the findings, we can conclude that the application created was able to successfully categorise the various types of Arrhythmia as well as determine its presence/absence with an accuracy of around 92%. In addition to this, the use of a Deep Convolutional Neural Network simplifies the learning process using the concept of weights and backpropogation.

## 10 FUTURE SCOPE

The proposed network, which is based on a convolutional network, is rather simple. With the usage of Inception models, it is anticipated that improved outcomes will be obtained. The proposed technique could be used in small devices that continuously monitor ECG signals, for example, to detect anomalies and establish an initial diagnosis, or to assist a doctor in this process.

## 11 BIBLIOGRAPHY

[1] Ge, D., Srinivasan, N., & Krishnan, S. M. (2002). Cardiac arrhythmia classification using autoregressive modeling. Biomedical engineering online, 1(1), 1-12.

[2] Song, M. H., Lee, J., Cho, S. P., Lee, K. J., & Yoo, S. K. (2005). Support vector machine based arrhythmia classification using reduced features.

[3] Tsipouras, M. G., Fotiadis, D. I., & Sideris, D. (2005). An arrhythmia classification system based on the RR-interval signal. Artificial intelligence in medicine, 33(3), 237-250.

[4] Wang, J. S., Chiang, W. C., Hsu, Y. L., & Yang, Y. T. C. (2013). ECG arrhythmia classification using a probabilistic neural network with a feature reduction method. Neurocomputing, 116, 38-45.

[5] Anuradha, B., & Reddy, V. C. (2008). CARDIAC ARRHYTHMIA CLASSIFICATION USING FUZZY CLASSIFIERS. Journal of Theoretical & Applied Information Technology, 4(4).

## APPENDIX A.

Github Repository Link: https://github.com/smartinternz02/SI-GuidedProject-49391-1652770520

Demo Video Link: https://youtu.be/Gfz_syQbI_M