IOT DEVICE TO DETECT ANEMIA

Project ID: 19-129

Individual Thesis report – Creation of diagnosing algorithm and the backend of the system

Comprehensive Design/Analysis Project

H.S.M.H Fernando IT16136120

B.Sc. (Hons) in Information Technology Specializing in Computer Systems and Network Engineering

Department of Information Systems Engineering

Sri Lanka Institute of Information Technology Sri Lanka

September, 2019

IOT DEVICE TO DETECT ANEMIA

Project ID:19-129

Creation of diagnosing algorithm and the backend of the system

Thesis report

(Thesis report submitted in partial fulfilment of the requirements for B.Sc. (Honors) degree in Information Technology Specialization in Computer Systems and Network Engineering)

H.S.M.H Fernando IT16136120

Supervisor Mrs. S. Lokuliyana

B.Sc. (Hons) in Information Technology Specializing in Computer Systems and Network Engineering

Department of Information Systems Engineering

Sri Lanka Institute of Information Technology

Sri Lanka

September, 2019

DECLARATION

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text. Also, I hereby grant to Sri Lanka Institute of Information Technology the non-exclusive right to reproduce and distribute my dissertation in whole or part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Date: Signature:

Date: Signature of Supervisor:

ABSTRACT

Health and safety is one of the key aspects which a person would be most concerned about. Diseases can have various adverse effects on human lives. Therefore, early detection of the disease is a key area that needs to be addressed. This group project is concerned with the diagnosis of anemia disease which is a disease caused by the lack of iron in the blood. This is a disease that would even lead to organ failure eventually leading to a heart attack or death. As the iron content in the blood of a person with anemia decreases, the colour of blood, skin and finger tips may seem less intense in red colour than a healthy person. The project is based on building a device that can read the red intensity of the skin colour of a person and using that reading along with other information gathered from the person via a questionnaire to produce a diagnosis of the person determining whether the person has anemia or not. This thesis report is concerned with the implementation of the machine learning algorithm used to produce the predictions and the backend infrastructure used to support the anemia detection system.

ACKNOWLEDGMENT

First of all I would like to thank Ms. Shashika Lokuliyana for her continued support and guidance given to the group in order for the successful implementation of the project. She as the supervisor for this project was very engaged from the off and the many fruitful discussions she had with the group helped immensely in the success of the project.

Next I would like to thank my fellow group members for their determination , drive and hard work in making this project a success. I received a lot of help from each of them in implementing my component of the project.

Finally I would like to thank my family for always being there for me and supporting me throughout this part of my life.

TABLE OF CONTENTS

LIST OF FIGURES

1 INTRODUCTION

1.1 Background Literature

Anemia is a disease that is caused due to the lack of iron in the blood which eventually reduces the blood hemoglobin level. This causes a lack of oxygen in the body. The ability to detect the disease at an early stage would lead to an early cure. The main method of detecting anemia requires drawing blood from a person. This can be dangerous especially in low income countries like Sri lanka where infections can happen due to this method. Also it is preferable to avoid drawing blood from a vulnerable person such as a pregnant mother or a child.

In order to achieve this, the team will be designing a non-invasive system which has the ability to detect anemia with a high accuracy rate and giving the output to the patient so that the treatments can be taken at an early stage. A device is built using a non-invasive method so that blood should not be taken and then the data is sent to the server for processing and the user has to fill the details in the app as well. Then the system able to analyse the data and get the results to the app.

In the invasive anemia detection methods, a device has been developed containing electronic instrumentalization, post processing software and plug and play disposable sensor which can be used to take 50µL whole blood sample to test for anaemia and provide instantaneous results [1]. Although proper testing has resulted in only 2% accuracy error , this being a noninvasive method does not solve our search problem.Non invasive devices like project that is done by R. Bhattacharyya which is using RFID the ability to test for anaemia [2] exists but the accuracy is not comparable to the traditional methods. This is why we are proposing a system where a device is present to take measurement from user and then that data is processed using image processing inside the device. Then that data is used together with user inputs through a questionnaire in the mobile app and all that data together is put through an algorithm to produce an outcoming determining whether the user has anemia or not. By doing research we can reasonably determine that a machine learning algorithm would be best suited for this type of work [3]. In a review by Mahmoud H. Qutqut et al [4], it is noticed that Artificial Neural Network (ANN) performed well in most models for predicting heart disease as well as Decision Tree (DT). As ANNs are more complex,resource hungry and require more data, I have chosen supervised machine learning algorithm type which is a classification algorithm. In implementing this component of the project I have chosen Machine Learning as a Service (MLaaS) and Backend as a Service (BaaS) as they provide many benefits [5] of traditional methods including costs saving, high scalability and reduced workload on the team.

Azure Machine Learning studio has been chosen as the MLaaS platform as it is a tool that allows a user to use data from one or more sources, transform, and analyze that data through various data manipulation and statistical functions, and generate a set of results in developing a predictive model [6]. Azure Machine Learning Studio allows to run experiments to create different predictive models and it is free to use with some limitations for research purposes. When deploying the anemia detection system as a commercial solution, Azure Machine Learning Studio offers commercial packages which can be utilized to scale the product as needed. Google Firebase service offers a platform to develop rapidly evolving application with a wide variety of services provided by Firebase [7]. I have used different features and service of Firebase in implementing the backend functionality of the anemia detection system.

1.2 Research Gap

As far as I am aware, there does not exist a non invasive anemia detection solution producing highly accurate results with the integration of a questionnaire. Hence there exists a gap in research where a solution achieving what we hope to achieve in this project is not present at this moment. A research gap exists regarding a research where the readings taken off of medical patients can be supplemented by an associated questionnaire to further enhance the final diagnosis produced by such a system.

Other projects on anemia detection work on a single input given to determine the result but to our knowledge there are no projects looking to take multiple input like in this project in order to produce results. There is also a research gap where a data set relating to the skin colour of patients is not available to be used for analysis. This research also hopes to fill that gap by producing a good data set so that future researchers can use it for their future work.

1.3 Research Problem

When developing the anemia detection system, my component is focused on the algorithm which will analyse the data received from the device and the mobile app to come up with the diagnosis and the implementation of the backend which will provide interfaces to the device, app as well as other necessary services. Although there exists Machine learning algorithms and other techniques (such as Neural Networks), most of them use only the data taken by a device to produce the results. But in this project, the team use multiple data inputs from the questionnaire in the mobile app in addition to the data gathered by the device to produce the result. This is unlike any other solution in the industry that is known. The data set used to train the prediction model produced by the machine learning model needs to be changed and adjusted in such a way that the best model possible is generated. As the data from image processing done at the device and the data from the mobile app is rather large in size and complexity it presents a significant challenge in obtaining an acceptable result at the completion of this component. With the scope and complexity of the input data there is also the challenge of adjusting the necessary thresholds and the parameters of the algorithm to minimize false positives and false negatives.

There also needs to a backend to handle all the data inputs and outputs from the device and the app and store them in an appropriate manner. The backend must also authenticate the device and user and must also provide management interfaces for management purposes. The challenge is there to come up with a solution that can handle the different formats of data to be exchanged with the MLaaS API and the mobile app API. The backend services must be available at all times and must be able to handle dynamic amounts of traffic. Finally the lack of suitable data sets in terms of images of skin is also a research problem. This means that in this project a completely new data set must be created and testing and validation must also be done using that data set rather than using existing verified data sets.

1.4 Research Objectives

The main goal of this project is to design a device so that the user is able to detect whether they are diagnosed with Anemia or not. It is important to accurately detect the presence of anemia at an earliest stage as possible. When a patient contracts anemia that person may experience many different symptoms. But one of the key symptoms that can be seen in an anemia infected patient is that the amount of iron reduces which leads to the reduction of oxygen in the blood, makes the blood look pale than red. This can be observed by taking images of the skin at fingertips. These images are sent through an image processing algorithm and the result will be sent to the backend of the system. Then that result will be combined with the results of the questionnaire from a mobile app and that data will then be sent to a machine learning algorithm which will then produce a diagnosis. The main objective of this machine learning algorithm is to create a highly accurate predictive model for predicting anemia diagnosis for users. Another objective is to create a proper dataset which can be used by other researchers in their future work.

2 METHODOLOGY

2.1 Methodology

In order to produce the final diagnosis we will be using Azure Machine Learning Studio to train a predictive model using the boosted decision jungle algorithm provided by Azure. A web service will be created using the predictive model which can be used to produce results. Azure Machine Learning Studio provides tools to do this. A datastore will be setup using Google Firebase so that the device and mobile users can be authenticated and their data be stored for use by the anemia detection system. The firebase project will be connected to the mobile application and this integration will allow for data storing and authentication. The result of the image processing will be uploaded to the datastore via a Python server running on the image processing platform and then once the corresponding data from the mobile app is uploaded, another Python server which monitors the state of the datastore will have a function triggered which will then send the data to the web service of the predictive model. Then the web service will produce a result which again once uploaded to the data store which will be notified to the user via the integrated mobile app.

The following steps were involved in implementing my component of the research project.

- Selecting suitable decision algorithm
- Integrating mobile app to backend
- Setting up python server
- Building dataset
- Training model
- Setting up web service for prediction model
- Testing

2.1.1 Selecting suitable decision algorithm

In this step several two class classification algorithms were compared using experiments in Azure Machine Learning Studio. The compared algorithms were boosted decision tree, boosted decision forest and boosted decision jungle. Experiments were run using a dataset that would be similar to our eventual dataset for this research with having 10 numeric valued attributes and with around 120 records. This is similar to our eventual dataset for this research so by running experiments on this dataset I was able to determine the best algorithm for this type of work by comparing the accuracy of each prediction model produced. The experiments run are shown in the figure 2.1.

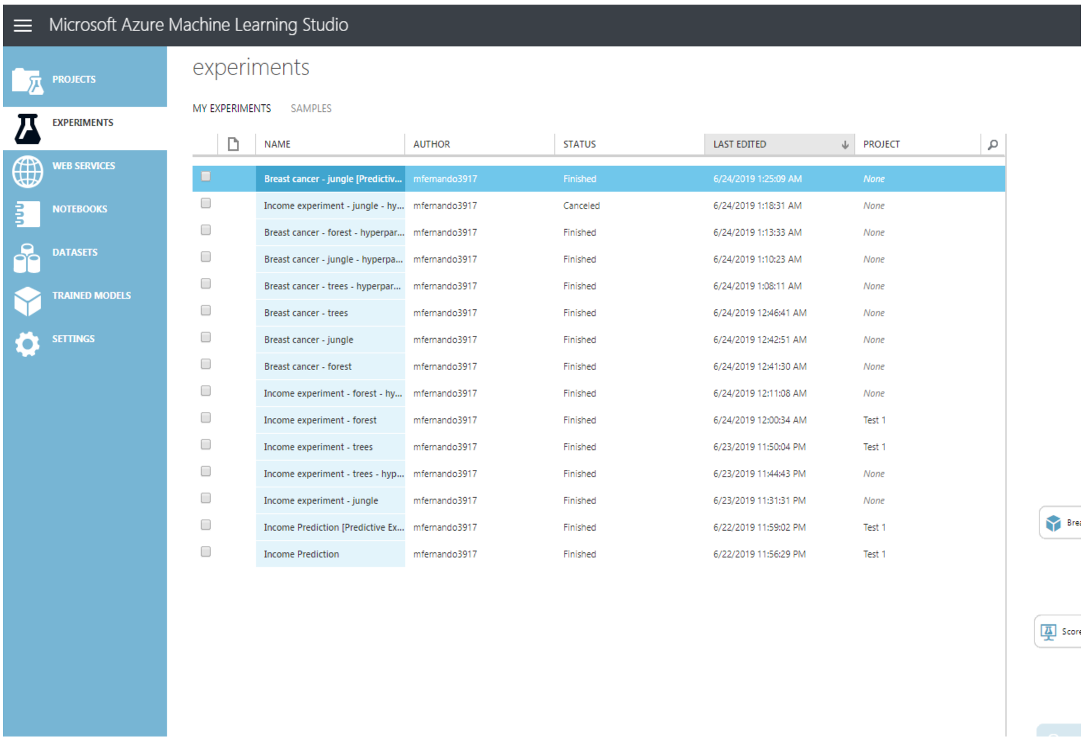


Figure 2.1 : Experiments

The best algorithm turned out to be two class boosted decision jungle algorithm for this type of work.

2.1.2 Integrating mobile app to backend

The mobile application was connected to the firebase project which is providing the backend functionality for the anemia detection system. Figure 2.2 shows the application connected to the firebase project after making the necessary configurations.

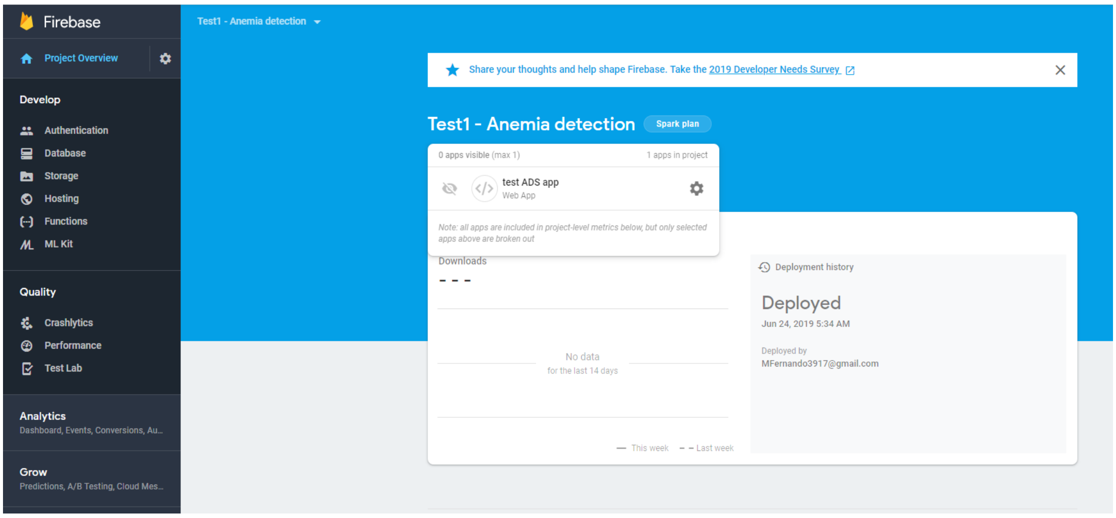


Figure 2.2 : Integration

The firebase cloud firestore service is used to create a database on the cloud so that data related to the system can be saved and authentication can be provided. A snapshot of the datastore is shown in figure 2.3.

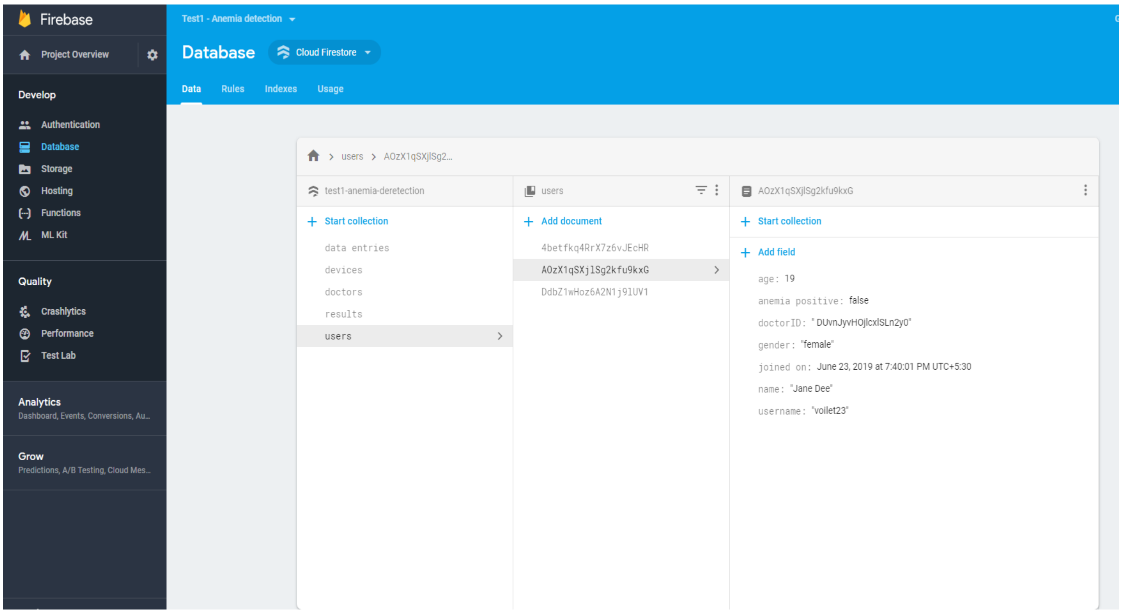


Figure 2.3 : Datastore

2.1.3 Setting up python server

A python server was setup to constantly monitor the datastore to detect when a user has uploaded data to be used in the machine learning web service, once detected to send the relevant data as a POST request to the web service and to store the result contained in the response to the POST request in the datastore. The portion of the server code used to compose the POST request sent to the machine learning web service and then to store the extracted result from the response to that request is shown in figure 2.4.

```python
data =  {

        "Inputs": {

                "input1":
                {
                    "ColumnNames": ["Healthy", "Q1", "Q2", "Q3", "Q4", "Q5"],
                    "Values": [ [ IPR, Q1val, Q2val, Q3val, Q4val, Q5val ], [ "0", "0", "0", "0", "0", "0" ], ]
                },        },
            "GlobalParameters": {
}
    }

body = str.encode(json.dumps(data))

url = 'https://ussouthcentral.services.azureml.net/workspaces/c194828abb25492faafef1241411b535/services/5b3b31de418d45a0872cd8f433085cdf/execute?api-versior
api_key = 'DTbhh434sD95RSAX/vlK/z70xgajuWTaCvVAFo6zqSq6Ej+mqQWgzY3Yic/jygNOQsn469JSvnEGuUh8XYuWQQ==' # Replace this with the API key for the web service
headers = {'Content-Type':'application/json', 'Authorization':('Bearer '+ api_key)}

req = urllib2.Request(url, body, headers)

try:
    response = urllib2.urlopen(req)

    # If you are using Python 3+, replace urllib2 with urllib.request in the above code:
    # req = urllib.request.Request(url, body, headers)
    # response = urllib.request.urlopen(req)

    result = response.read()
    print(result)
    print(result[131])
    score_label = result[131]
    score_label_value=int(score_label)
    print(score_label)
    score_probability = result[135]
    score_probability_value=int(score_probability)
    print(result[135])
    print(score_probability)

    doc_score_ref = store.collection(u'scored_predictions').document(u'IpHRjF8jqQnrC6Qj0md4')
    try:
        doc_score_ref.update({u'scored_label': score_label_value})

    except google.cloud.exceptions.NotFound:
        print(u'Missing data')
    try:
        doc_score_ref.update({u'scored_probability': score_probability_value})
```

Figure 2.4 : Python server code

The python server is a continuous while loop which checks the datastore for changes and triggers a series of functions when changes are detected as shown in figure 2.5 which contains the console log of the server.

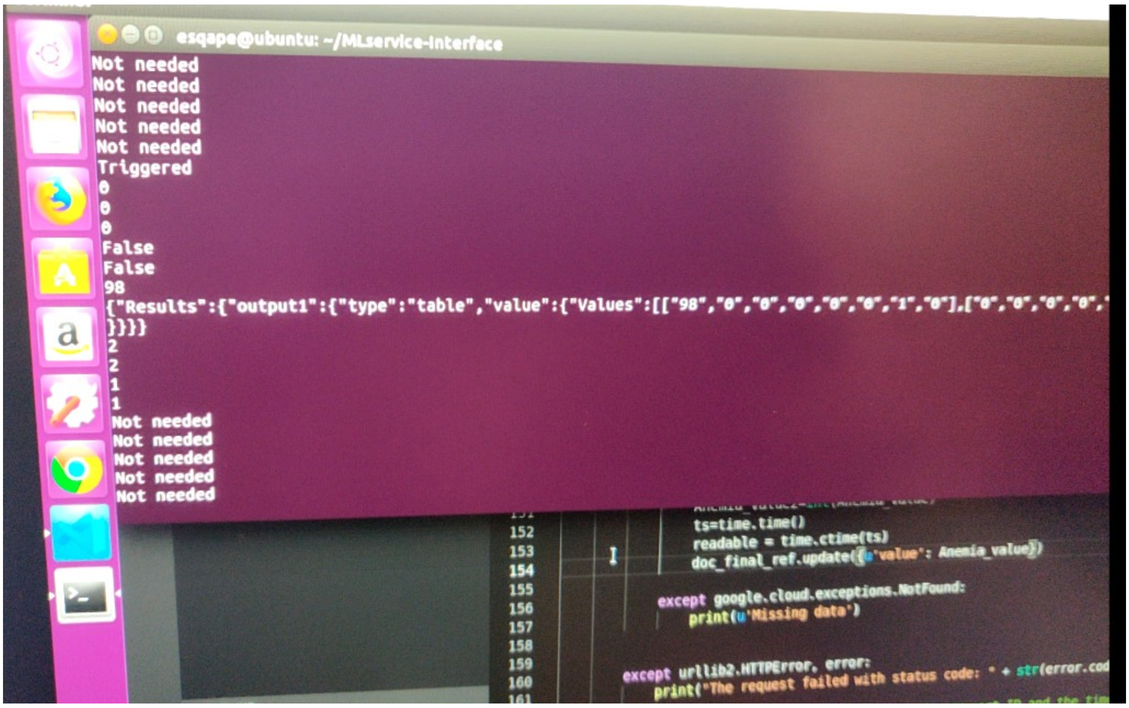
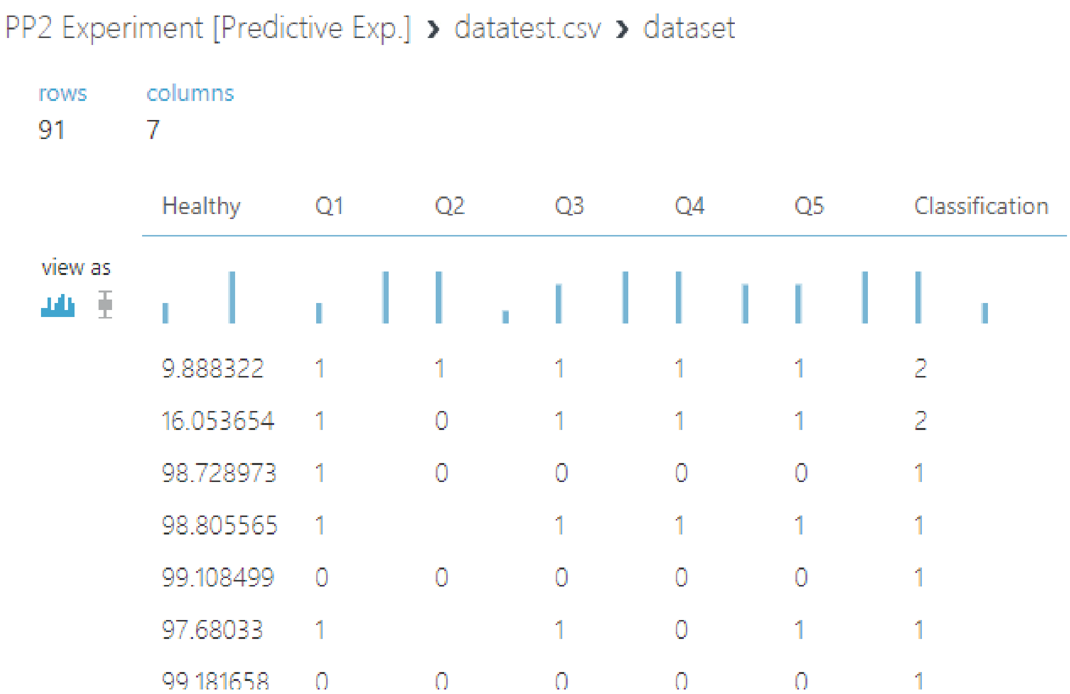


Figure 2.5 : Python server in action

2.1.3 Building dataset

The data from the image processing algorithm and the mobile application questionnaire must be collected and then transformed in various ways in order to build a data set that can yield an accurate predictive model when put through the learning algorithm. Qualitative data must be formatted to qualitative, numeric data and for each variable the weight of the variable must be determined so that a desirable prediction model is produced by the learning algorithm. Over 105 data samples were collected from which only 91 were used in the final dataset used to train the predictive model after applying data cleaning and transforming techniques. The final dataset was then composed in .csv format as shown in figure 2.6.


PP2 Experiment [Predictive Exp.] › datatest.csv › dataset

rows 91 columns 7

Healthy	Q1	Q2	Q3	Q4	Q5	Classification
9.888322	1	1	1	1	1	2
16.053654	1	0	1	1	1	2
98.728973	1	0	0	0	0	1
98.805565	1		1	1	1	1
99.108499	0	0	0	0	0	1
97.68033	1		1	0	1	1
99.181658	0	0	0	0	0	1

Figure 2.6 : The dataset

As can be seen in figure 2.6 above, 91 records were present in the dataset with 7 columns. The first column represents the healthy classification percentage generated from the image processing algorithm for each user. The next columns from Q1-Q5 represents the answers to the questionnaire provided by each respective user with 0 representing false and 1 representing 2. The final column labeled Classification is the actual anemic diagnosis for each user with 1 representing a healthy status and 2 representing anemic status.

2.1.4 Training model

After building the dataset, the prediction model must be trained until a prediction model with desirable accuracy levels is produced. Figure 2.7 shows the training completed for the prediction model.

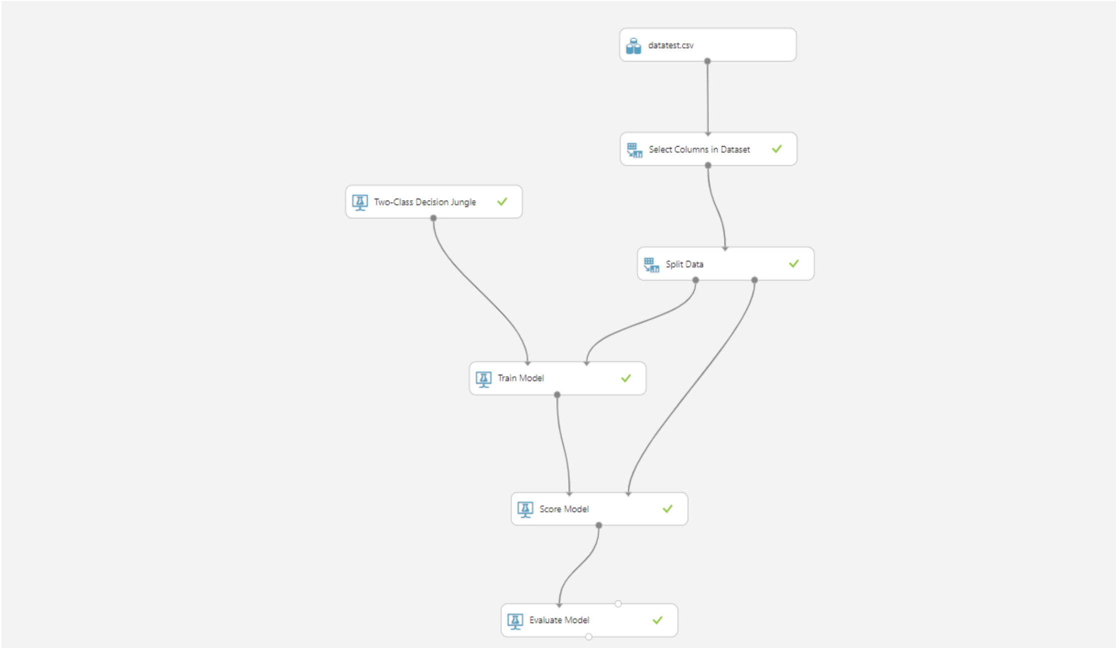


Figure 2.7 : Training

Here we are aiming for a prediction model that can predict the anemia condition of a person at an accuracy of 80-90 %.

2.1.5 Setting up web service for prediction model

After obtaining an adequate prediction model it is then necessary to implement the prediction model as a web service so that the backend can send API requests and obtain results for patients. Figure 2.8 shows the implementation of the prediction models used as a web service through Azure Machine Learning Studio.

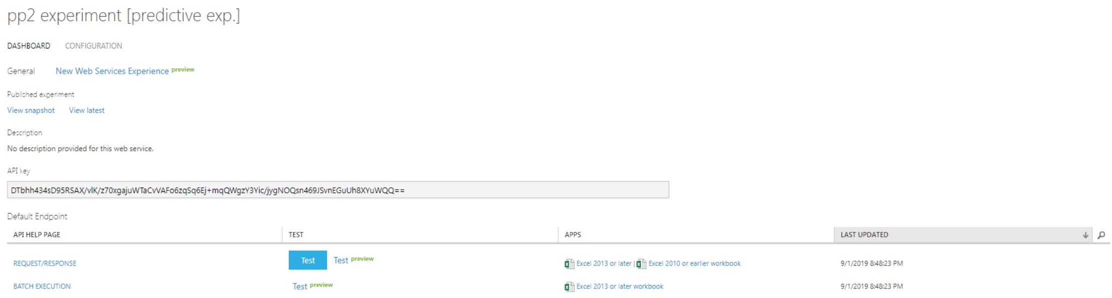


Figure 2.8 : Web service

As can be seen an API key is provided for each web service so that requests can be made to the service and obtain the results form the prediction model.

2.1.5 Testing

Testing will be carried out by using a portion of the dataset as shown in the figure 2.9.

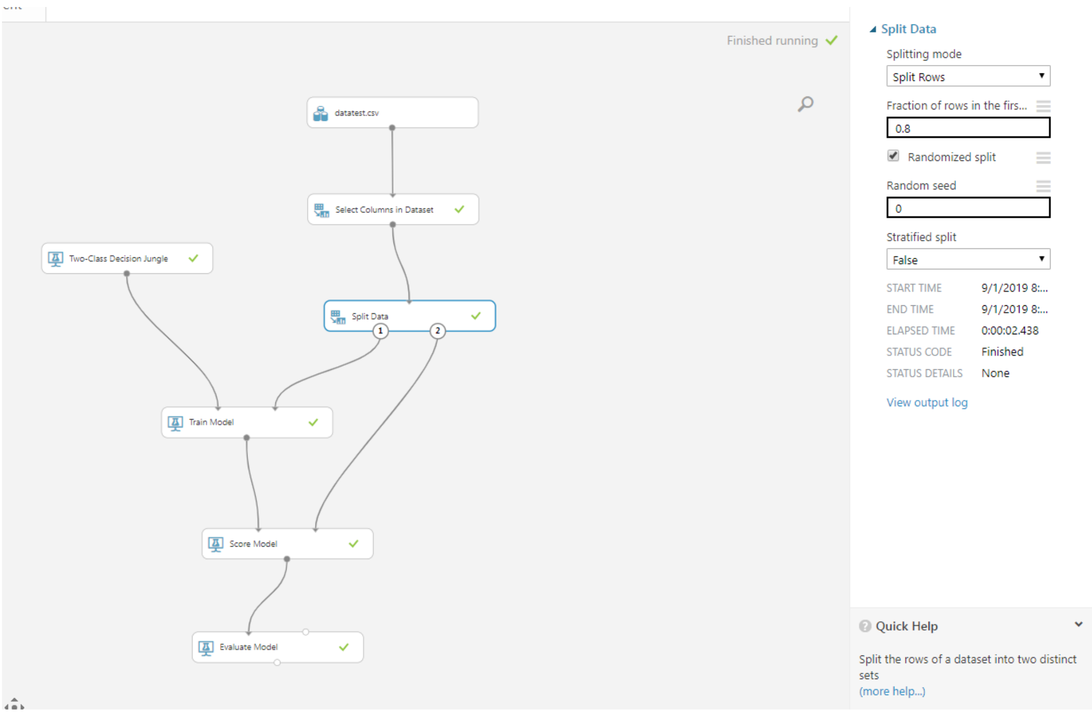


Figure 2.9 : Testing

As shown in figure 2.9 above, the dataset will be split into two portions and one of the potions will be used for testing of the prediction model produced by consuming the other portion. 80% of the dataset is used to train the predictive model while the other 20% is reserved to test the accuracy of the model.

2.2 Commercialization aspect of the product

As anemia is a widespread disease there is demand for a detection system especially for non invasive systems. The device is designed to be used by many users. The service based products used to implement the system means that the system can be scaled efficiently and effectively with increasing demand. Applications from other platforms such as web applications and iOS applications can also be connected to the same firebase project so this means that if the need arises due to commercial demands, those applications can be integrated very easily. Both Azure and Google offer reasonable service packages as the need to scale the system increases. Also with the experience and knowledge gained from this research, a mobile application that detects anemia using the phone camera can be developed which could be a huge commercial success.

2.3 Testing and Implementation

Testing should be done by corroborating the results with results taken from a sample of anemia patients. Accuracy of the diagnosis as well as other functionality must be ensured. First stage of testing will comprise of using the split data from the dataset to immediately determine the accuracy of the system. Then further testing must be done by using the system to test another sample of users and comparing the results against blood tests doe for those same users. During testing there can be setbacks due to the bugs and errors. But the fixes should also be done as soon as possible and make sure the device is functional and also be a guarantee that the information that is obtained taken through the device is also kept securely because the medical information are sensitive information therefore the proper security needs to be provided.

3 RESULTS AND DISCUSSION

3.1 Result

Inorder to assess the suitability of the predictive model for the system, the data split from the dataset for testing can be used against the produced predictive model after training the model. Figure 3.1 below shows the evaluation results of the split test data against the predictive model.

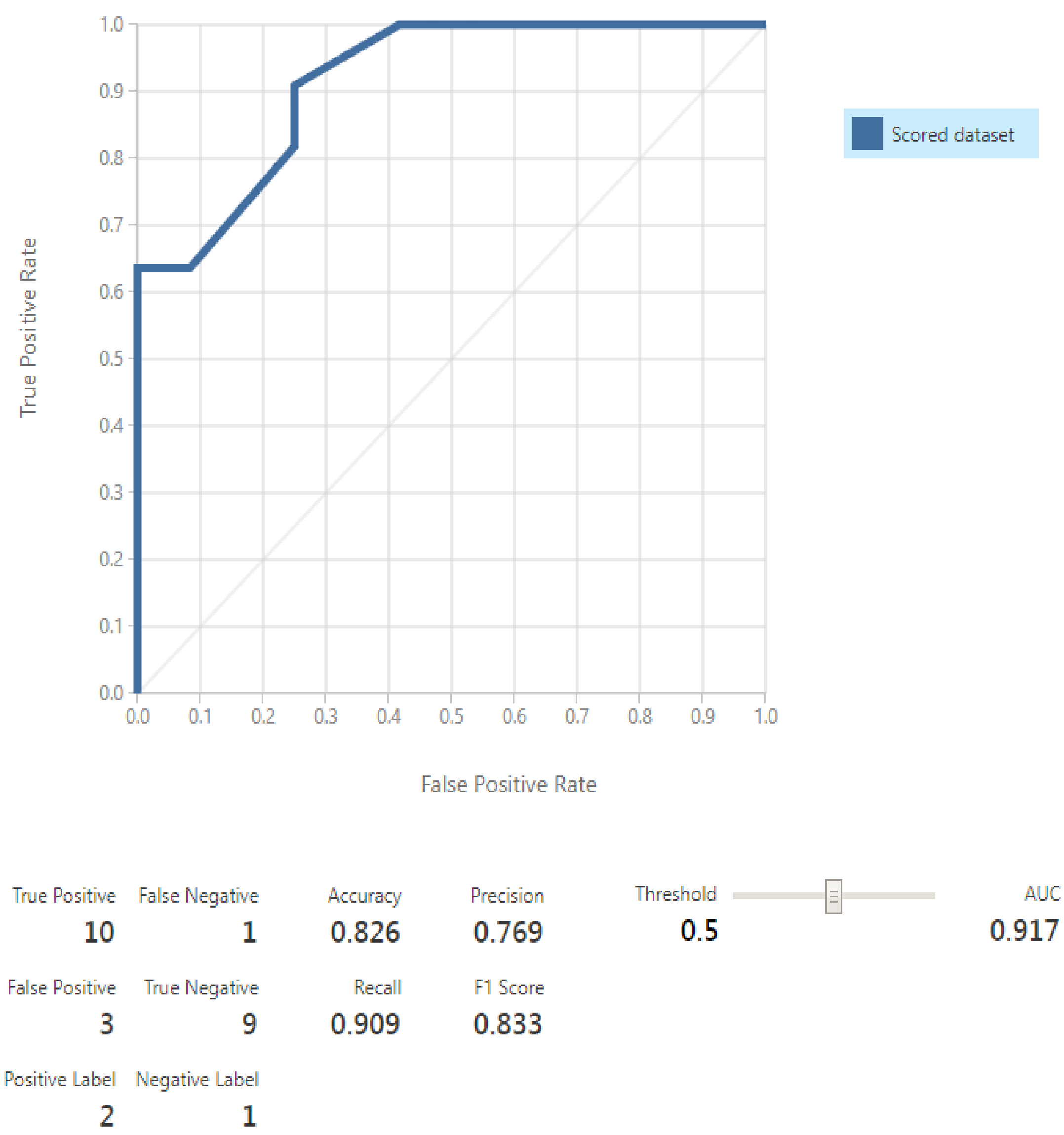


Figure 3.1 : Predictive model evaluation

As shown in the evaluation results, the accuracy of the predictive model is found to be around 82.6%. This accuracy has been achieved by using a two class decision jungle machine learning algorithm to tain the predictive model. As such, the system is currently able to predict the anemia diagnosis of a user with an average accuracy of 82.6%.

3.2 Research Findings

The main obstacle in completing this research was the collection of data samples. During this research there were certain constraints that can be faced as mainly it is the lack of dataset availability that would limit the machine learning algorithm from producing a more accurate and precise predictive model. As the dataset availability is low, the necessary had to be manually collected by visiting hospitals and making sure the appropriate patients are selected and the device developed for the project is used to collect the data samples that is used to train the model or else the accuracy level would be different if the samples are taken using some other resources which could alter the accuracy of the final results. Therefore,the building of the dataset by collecting data from scratch was an important aspect of the research. The proper organization of the data collected is also an important factor to note during the training of the predictive model.The collected data must be cleaned and transformed into the comma separated values format with each column labeled properly. During the data collection, the information collected should also be kept safe as the data are sensitive data and that information should be kept confidentially. If the server is hosted in the cloud, then a cost factor would be attached and in order to reduce it the services can be implemented locally in a server instance.

Another challenge faced was selecting the appropriate machine learning algorithm among the many algorithms and techniques which exist to classify data as needed in this research. Although there exists algorithms such as Artificial Neural Networks which can predict with extreme accuracy, those models require datasets with thousands of records. With the dataset for this research being limited to around 100 records, this posed a challenge in selecting a suitable algorithm which will produce results with acceptable accuracy. Thorough research and testing was made to finally settle on the decision jungle algorithm used in this research. This algorithm proved to be capable of producing acceptably accurate results with the limited dataset that was present.

It was also found that using a dedicated python server on a local machine was more efficient and much faster than using some other service such as firebase functions to connect the datastore with the web service of the machine learning predictive model. If using firebase functions, the time taken to gather the data from the datastore and to retrieve the result from the machine learning web service was significantly greater than using a custom python server to perform the same duties. This also meant that resources could be allocated to the python server as needed without having to consult third party providers.

3.3 Discussion

In this research, for the implementation of the machine learning solution, two class decision jungle algorithm by Microsoft was used to train a predictive model which was then set up as a web service so that data can be taken from the datastore and the results be stored back in the data store. The dataset used to train the final model was built by taking the image processing result and the answers from the mobile app questionnaire and transforming that data into comma separated values format with numeric values representing each data. The selected algorithm proved to be suitable for the research with the prediction model scoring an accuracy of 82.6% with the portion of the dataset itself used to test the accuracy. A custom Python server was coded with its functionality being to monitor the firebase datastore for changes and to take the relevant data to send a POST request to the predictive model web service and then to store the result in the datastore. The firebase datastore was set up to store the necessary data needed for the system to function along with direct integration to the mobile application.

4 CONCLUSION

In conclusion the system produced by this research is able to predict the anemia condition of a user with acceptable accuracy. The dataset created by combining the image processing result and the answers to the questionnaire was used to yield a predictive model through machine learning which could predict the anemic status of a user. This system is up to a certain level, suitable for pre diagnosis of anemia without the need of blood testing. There is great difficulty in creating dataset for this type of research and the produced dataset in somewhat limited in size. With more time and effort , a more extensive and better dataset could be produced which would yield better accuracy in the predictions. Also with a more extensive dataset, algorithms such as Artificial Neural Networks can be looked at to produce an even better predictive model. In the end, the research objectives for this component as well as the overall research were achieved with reasonable results. There is optimism that this research and the dataset produced would inspire more research into the field of health informatics and thus help in improving lives all over the world.

5 REFERENCES

[1] J. C. C.-F. R.-V. L. M.-C. Jaime Punter-Villagrasa, "Toward an Anaemia Early Detection Device Based on 50-μL Whole Blood Sample," IEEE, Madrid, 2015.

[2] R. Bhattacharyya *et al*., "Towards low-cost, wireless blood anomaly sensing: An RFID-based anemia detection sensor," *2015 IEEE International Conference on RFID (RFID)*, San Diego, CA, 2015, pp. 189-196.

doi: 10.1109/RFID.2015.7113091

[3] M. P. Meherwar Fatima, "Survey of Machine Learning Algorithms for Disease Diagnostic," Scientific Research Publishing Inc, Multan, Pakistan , 2017.

[4] Aljanabi, Maryam & Qutqut, Mahmoud & Hijjawi, Mohammad. (2018). Machine Learning Classification Techniques for Heart Disease Prediction: A Review. International Journal of Engineering and Technology. 7. 5373-5379. 10.14419/ijet.v7i4.28646.

[5] B. Schoepfle, "How "as a Service" (aaS) Model Works & How Companies Benefit from It", iT1 Source, 2019.

[6] "What is - Azure Machine Learning Studio", *Docs.microsoft.com*, 2019. [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/studio/what-is-ml-studio. [Accessed: 12- Aug- 2019].

[7] "Cloud Firestore | Firebase", *Firebase*, 2019. [Online]. Available: https://firebase.google.com/docs/firestore. [Accessed: 12- Aug- 2019].