**CHAPTER FOUR**

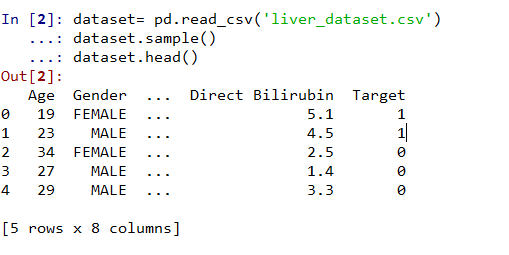
**IMPLEMENTATION, RESULTS AND DISCUSSION**

**4.1 System Implementation**

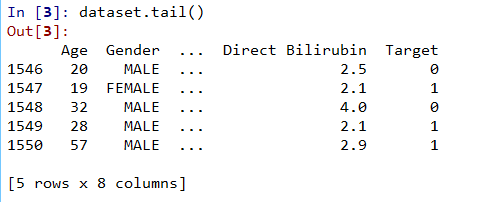
System implementation demonstrates the procedures and actions used to create the system. The implementation of the Liver Disease Prediction System is divided into two major parts, the User interface development and the Prediction Model development **using Anaconda Spyder Python packages and libraries** (Pandas, Numpy, Sci-kit Learn, Matplotlib, Seaborn).

**Dataset**

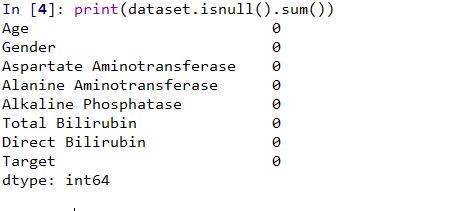
The dataset (Liver Function Test) used for this research work was collected in form of hardcopy from Federal Medical Center (FMC), Lokoja Kogi State. The collected hardcopy dataset was snapped and retyped in Microsoft Excel Spreadsheet format with .csv file extension. During the retyping of dataset, some predictors/observations with missing values were skipped, this is to ensure efficient data cleaning and preprocessing phase. Eventually, the retyped dataset used constituted seven (7) features **namely** Age, Gender, Aspartate Aminotransferase, Alanine Aminotransferase, Alkaline Phosphatase **respectively and** 1551 observations. However, other features/predictors of liver diseases such as Albumin and Total Proteins were not available in the raw data collected hence, they are not included in the dataset. **The dataset in the proportion of 75% to 25% was used for training and testing of the model respectively. Below in Figure 4.1 and Figure 4.2 are snippets of the dataset used for the prediction.**



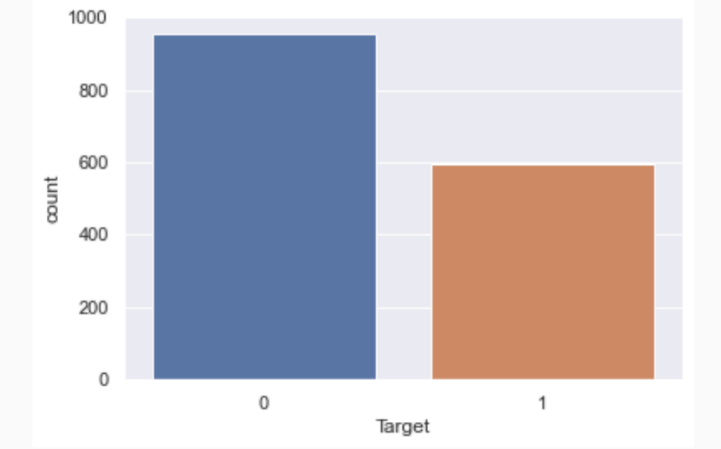
*Figure 4.1: Dataset sample.*

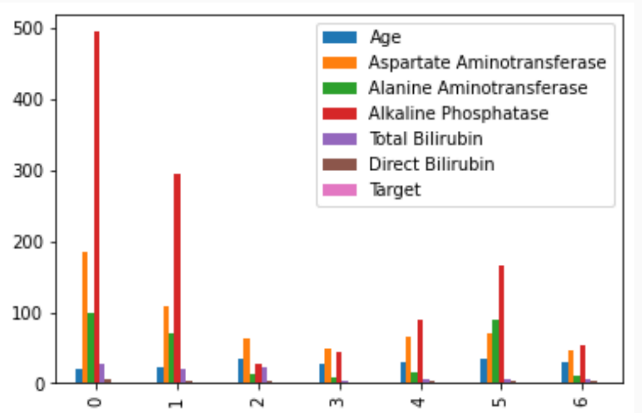


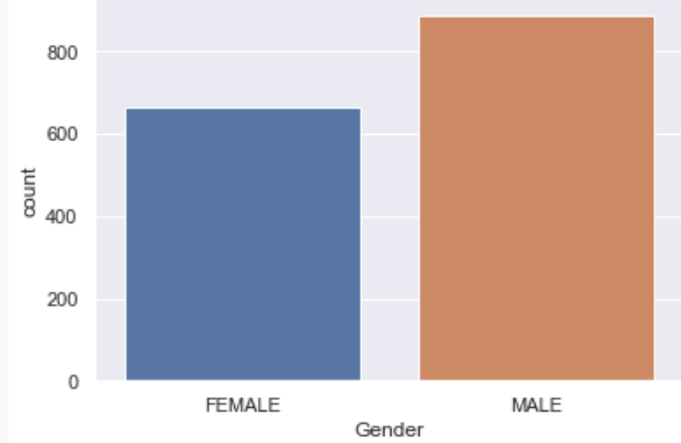
**Modelling**

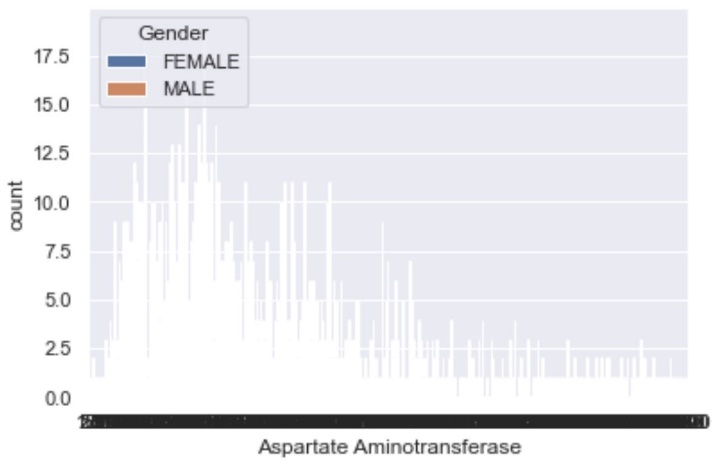
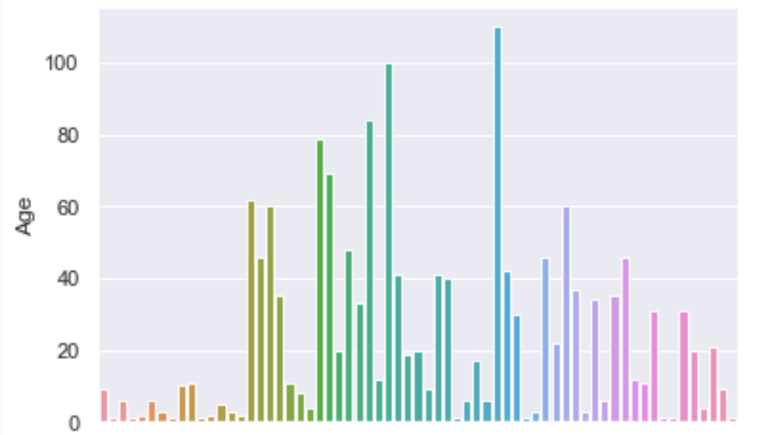
****

*Figure 4.3: Checking for missing value and dataset description*



****

****

****

*Figure 4.3: visualizing the Exploratory Data Analysis*

**4.1.1 Program Modules Specification**

This prediction model is made up of three different main functional modules:

1. **User Information Module**: This is the first module the user interacts with. It begins the process of the liver disease prediction system. Here the user input his/her blood test result report.
2. **Liver disease Prediction Module**: This the module that predict if the person has liver disease or not.
3. **Result Display Module:** This module ends the activity of the system by give the user a feedback base on the given input.

**4.1.2 Implementation Tools Used**

The tools used during the implementation phases or processes.

1. Anaconda IDE
2. Spyder Python 3
3. Python Shell
4. Visual Studio Code

**4.1.3 Development Process of Liver Disease Prediction Model/System**

The system development process started with the planning and analysis phase, the high rate of Death lead to the development of a system that can predict if a person has liver disease or not. Current system was observed, existing document were analyzed and requirement gathering took place. These project follows the Agile software development process model. The model offers the ability to develop and make changes to the system while still in the process. Changes to the system can be affected as many times as possible before final deployment.

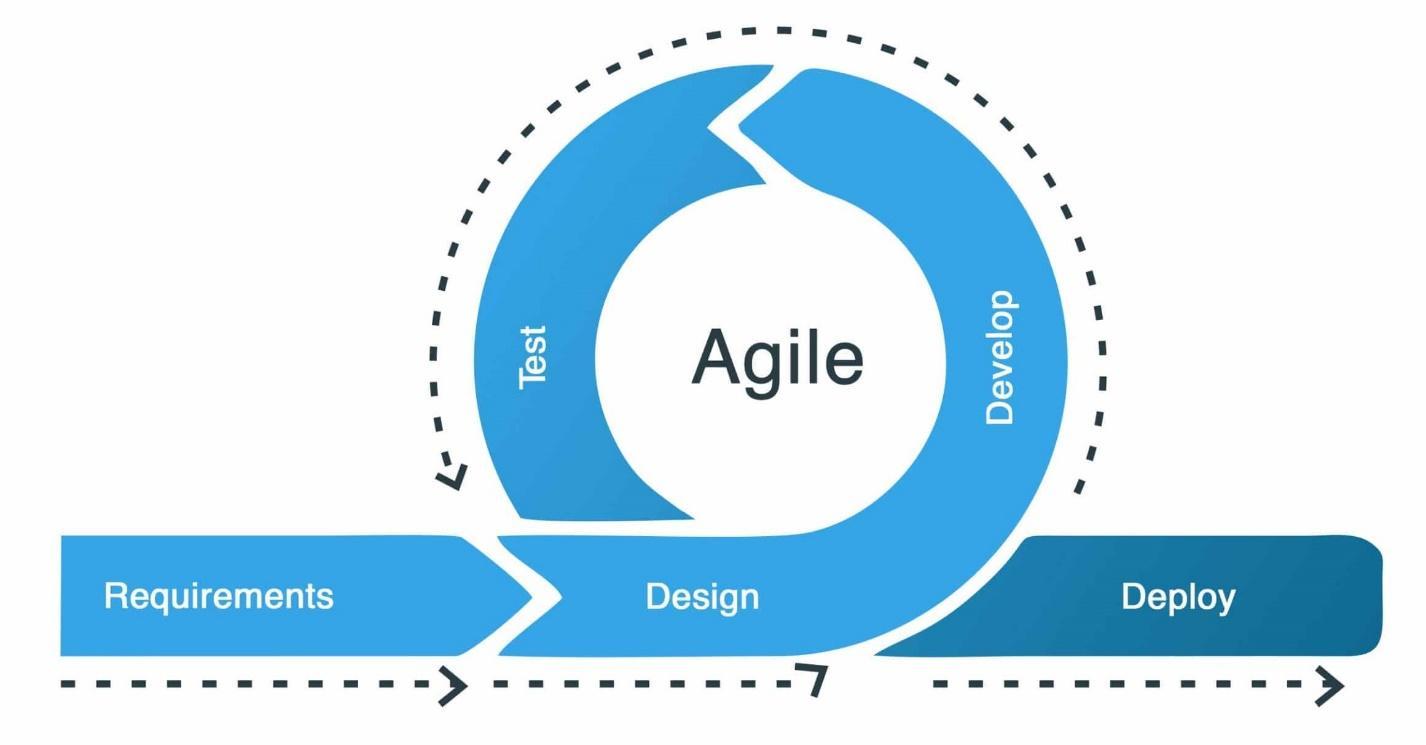


Figure 4.4**:** Agile System Development Process Model

Source:*<https://joinsoft.com/wp-content/uploads/2021/06/07-Agile-Software-Development-Process-Model-in-Software-Engineering-scaled.jpg>*

* Requirement: This is the first stage of the agile process model. it deals with information gathering on the needed tools for the development of the system.
* Design: This deals with creating a design of the system following the requirements for the development.
* Develop: The design produced from the second step above is converted into a real system using codes.
* Test: At this stage, it is ensured that the software is without error and compatible with everything that was designed in previous stages.
* Deploy: Deployment is the final step in the process model; this is the stage where the system is put into use.

**4.1.4 Physical Implementation/Deployment**

The implementation was done using python as the programing language and Spyder as the IDE which the model was finally save as pickle File, then deploy on a local host with the use of Python Flask.

The Physical deployment of this project involved the following steps:

1. Environmental Setup: The enabling environment for the local host or deployment of liver disease prediction model was set up using Anaconda Command Prompt by installing python flask and other necessary libraries.
2. Web Application Development: This involves building or design the necessary front-end web application that provides an interface for users to input his blood test report details or features and retrieve the predicted result.
3. Model Deployment: This is the process of deploying or transferring the model file to the executable environment after the environmental setup, for the model to be host locally.
4. Testing and Validation: This deals with conducting thorough functional and performance testing of the local deployed model to ensure that it functions correctly and provides accurate prediction of liver disease.

**4.1.5 System Requirements**

System requirements are the minimum or maximum hardware and software specifications that a system must meet in order to function properly. It is also the configuration or the required specifications that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result performance problems.

**4.1.5.1 Hardware Requirements**

The minimum hardware requirements to run liver disease prediction model are as follows:

1. 20 Gigabyte free Hard Disk Drive
2. 4 Gigabyte RAM

**4.1.5.2 Software Requirements**

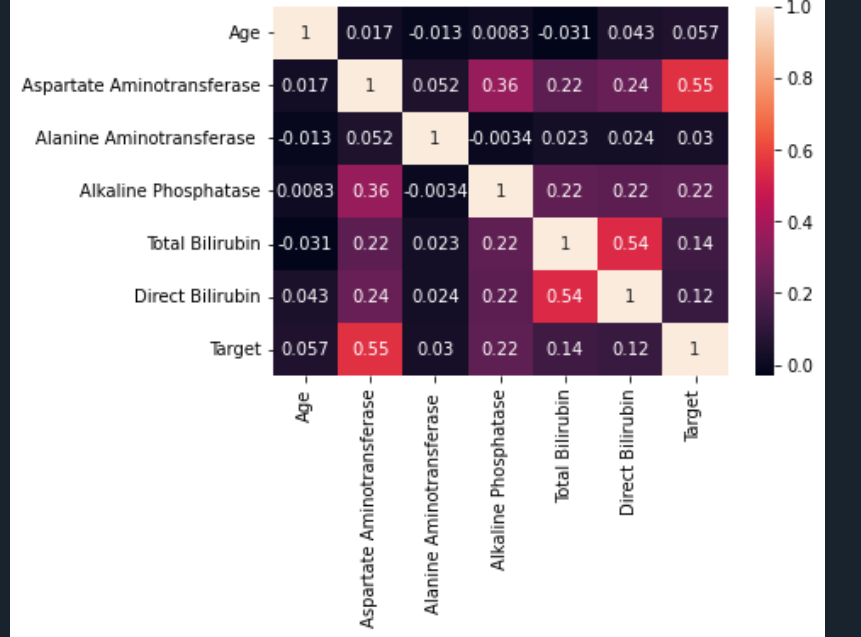
Below are the minimum software requirements to run liver disease prediction model

1. Spyder Python 3
2. Windows 8 and above, Operating System
3. Anaconda IDE
4. Python Shell
5. Web Browser

**4.2 Presentation of Results and Interpretation**

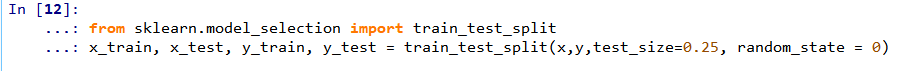
This section deals with the presentation and interpretation of the findings and insights derived from the development of the model and its predictions. Below are some results and interpretation:

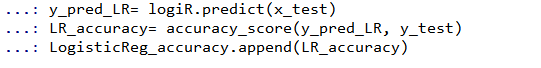
**Correlation**: It is the visual representation of data fitness which demonstrate a relationship between variable. However, the correlation in this research project indicated Perfect Positive Correction (1) which means that the variable tends to move in the same direction. That is, when one variable increase, the other variable also increases.



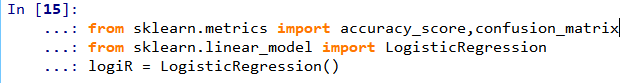
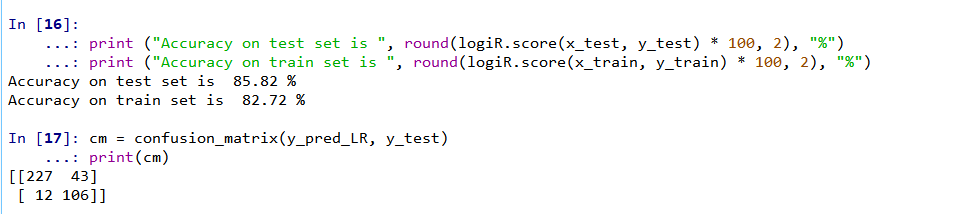
*Figure 4.5: Model Performance by Correlation*

**LOGISTIC REGRESSION**

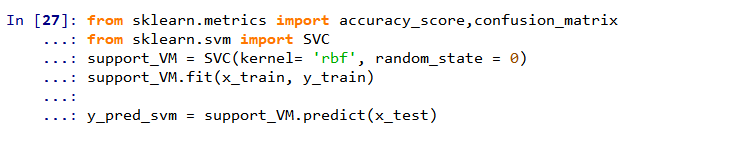
****

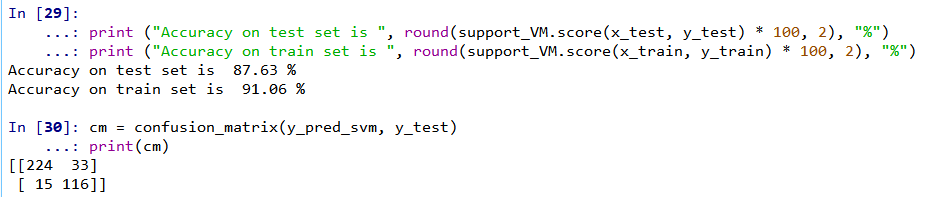


**LOGISTICS REGRESSION EVALUATION**

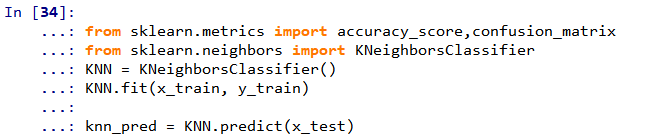
****

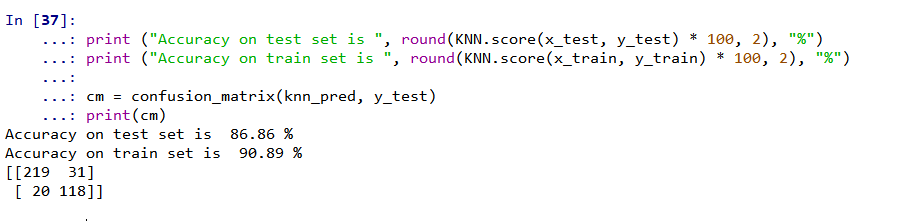
**SUPPORT VECTOR MACHINE**

****

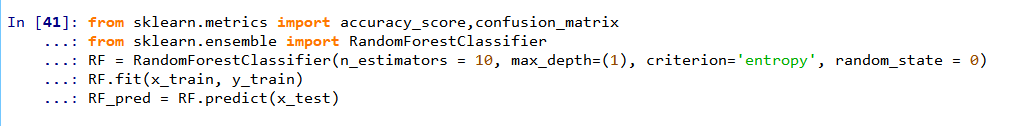
****

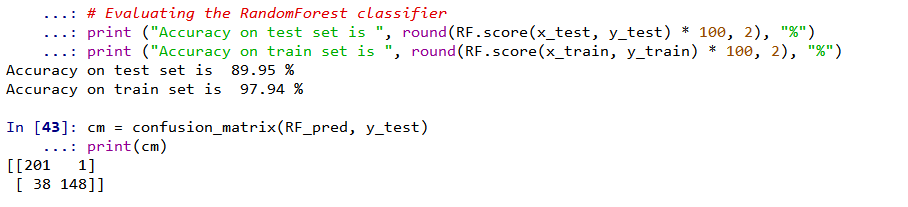
**KNEIGHBOR CLASSIFIER**

****

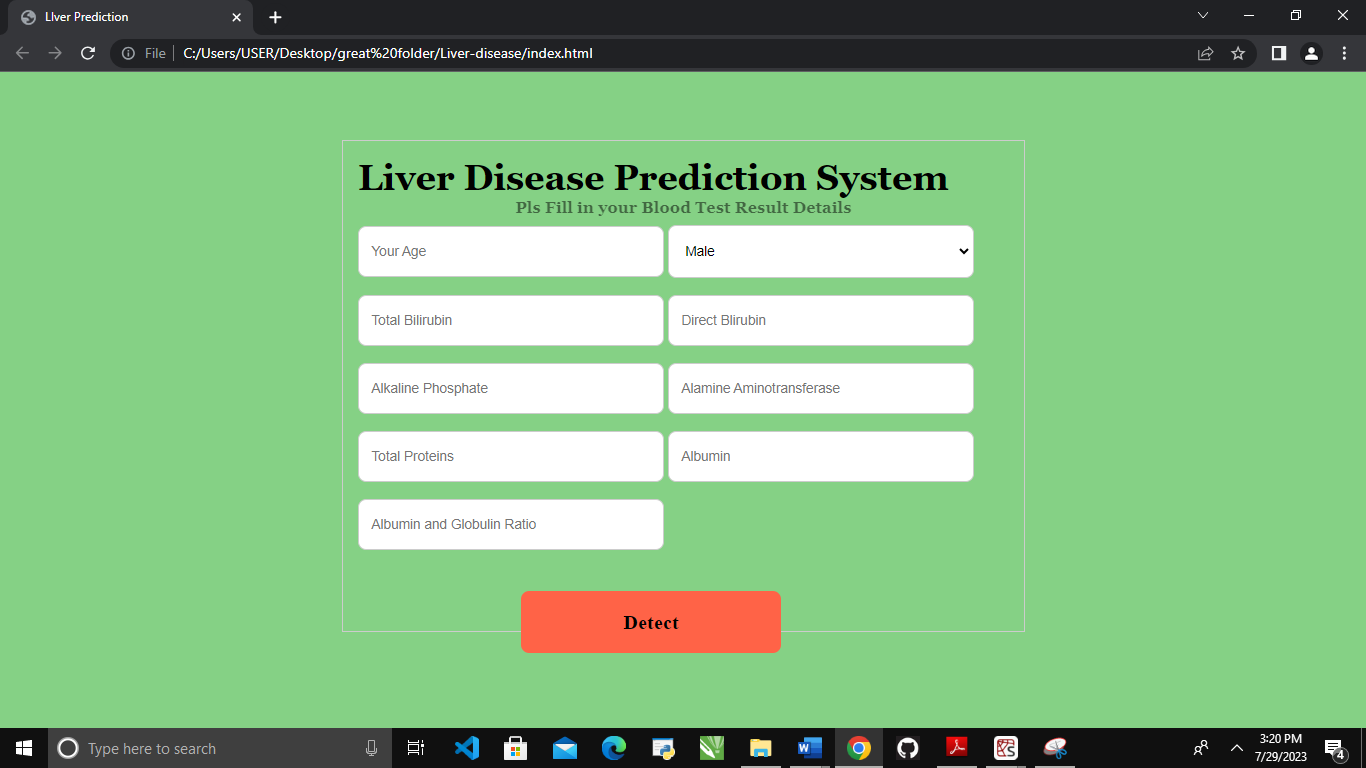
****

**RANDOM FOREST CLASSIFIER**

****

****

**4.3 Discussion of Results**



*Figure 4.6: Result 1*

Figure 4.6 shows how the user interface presents the outcomes of testing the liver disease prediction model on a variety of blood test report gotten.

The model accuracy was also checked using confusion matrix and the graphical representation is attached to the appendices of this project also cross validation was also use to validate the result from this model’s accuracy score, data pruning tuning was used to reduce overfitting in the model.

**4.3.1 Performance Evaluation**

The performance evaluation of the system was carried out using different metrics to determine the efficiency of the system, and also to know how accurate the system can predict liver disease to patient. The metrics used to evaluate the performance of the prediction model are confusion matrix, accuracy score, F1 score and Cohen Kappa Score. Below is the performance evaluation for the four algorithms implemented in this research.

Table 4.1: Model Performance Evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| MODEL | ACCURACY\_SCORE | F1 SCORE | COHEN KAPPA SCORE |
| Logistics Regression | 85.82% | 0.86 | 0.68 |
| Support Vector Machine | 87.63% | 0.88 | 0.73 |
| K-Nearest neighbor | 86.86% | 0.87 | 0.72 |
| Random Forest Classifier | 89.95% | 0.90 | 0.80 |

From the result depicted above, Random Forest classifier yielded the best result with an accuracy of 89.95%. Therefore, Random Forest was used for modeling.

**4.3.2 System Testing**

System testing is the examination of a fully integrated software solution. Since it does not

call for understanding of the internal structure of the code, this kind of testing is referred

to as "black-box testing." It ensure that the code is error free, produce correct output, reliable and efficient. Both functional and non-functional requirement should be considered. Testing was done for this system at several levels:

1. **Unit testing**: was done to test each component of the computer code while the

system was being developed. In the case of this project, all the modules

were tested independently and they worked well.

1. **Usability testing**: Usability testing was carried out to evaluate how well users

interacted with the system.

1. **Functional testing**: This was done to make sure the system operates and functions

as the user intended.

**4.3.3 Changeover Procedures**

The model was first trained on the raw dataset but the performance was overfitting with 100% accuracy on the trained test using Random Forest classifier, data pruning and tuning was considered in order to remove overfitting then the model was retrained and finally arrive at the above accuracy.

**CHAPTER FIVE**

**SUMMARY, RECOMMENDATIONS AND CONCLUSION**

**5.1 Summary**

Predicting liver disease at the early stages is a crucial step in increasing the survival rate of liver disease patients before it becomes severe. Because Liver is a complex disease, diagnosing it is difficult. A way to improve the disease procedures is needed in the medical industry. To determine a diagnosis, a physician must examine the results of a patient's test and compare them to those of other patients in comparable situations to assess prior judgements and because it is subject to the doctor's interpretation, the analysis of the factors that influence the diagnosis can be altered by human error (Viloria *et al.,* 2020). With the development of the Liver Disease Prediction System in this project will solve the issue of validity of diagnosis and time-consuming factor.

**5.1.1 Review of Contributions and Achievements**

Diagnosing Liver disease would become better and more accurate using the Liver Disease Prediction System developed in this project. Following the manual process of getting a diagnosis is less accurate and also time consuming.

**5.2 Recommendations**

This Prediction system by its advantage of enhancing the validity of the conclusiveness of a Liver disease diagnosis is highly recommended for hospitals, clinics, diagnostic center and health sector in general.

**5.2.1 Suggested Areas for Future Research**

Future work on this research project should be on the improvement of the model to be able to predict/diagnose the specific type of liver problem and guideline on how to abstain from toxic things that will further damage the liver.

**5.3 Conclusion**

A reliable and time-saving Liver disease prediction System was developed in the course of this research project. The model consists of two major parts, the user interface designed with html, CSS and the machine learning algorithm with python flask which serves as the backend for the model. Based on algorithm applied, it is observed that Random Forest algorithm gives better accuracy than the other classification algorithm. The interface enhances easy and smooth interaction between the user and the model; these machine learning model having an accuracy of 89.95% can be reliable for predicting liver disease.