**Software Requirements Specification**

**for**

Predicting Spine Abnormalities using Machine Learning

**Version 1.0 approved**

**Prepared by**

**Biswajeet Sahoo(1705689)**

**Daibik Dasgupta(1705692)**

**Harshit Agarwal(1705697)**

**Kush Jayank Pandya(1705701)**

**KIIT Deemed To be University**

**24 - April - 2020**

**Table of Contents**

**Table of Contents ii**

**1. Introduction 1**

1.1 Abstract 3

1.2 purpose 3

1.3 Document Conventions 3

1.4 Intended Audience and Reading Suggestions 3

1.5 Product Scope 4

1.6 References 4

**2. Overall Description 2**

2.1 Product Perspective 4

2.2 Product Functions 4

2.3 Technology Used 4

2.4 Operating Environment 5

2.5 Design and Implementation Constraints 5

2.6 User Documentation 5

2.7 Assumptions and Dependencies 5

**3. External Interface Requirements 3**

3.1 User Interfaces 5

3.2 Hardware Interfaces 5

**4. System Features 4**

4.1 Highly Accurate ML model 6

4.2 A Decent GUI 6

**5. Other Nonfunctional Requirements 4**

5.1 Help Page 6

**6. Future Enhancement**

Appendix: Analysis Models 7

# In**troduction**

## Abstract

*Lower back pain (LBP) is caused because of assorted reasons involving body parts such as the interconnected network of spinal cord, nerves, bones, discs or tendons in the lumbar spine. LBP is pain, muscle pressure or stiffness localized underneath the costal edge or more the substandard gluteal folds, with or without leg torment for the most part sciatica, and is characterized as endless when it holds on for 12 weeks or more then again, non-particular LBP is tormented not credited to an unmistakable pathology such as infection, tumour, osteoporosis, rheumatoid arthritis, fracture, or inflammation. Hence aiming to look for preventive measure rather than curative, this study suggests a classification methodology for Chronic LBP disorder using Machine Learning techniques.*

*We are going to have an exploration of this data and then predict lower back pain using these biometric measurements; this is a binary classification problem, as we have two outcomes: abnormal and normal. There are many ways that we can approach a classification problem, and there are many algorithms that can be used to create predictions. In this report, we use logistic regression, random forests and TensorFlow Lib. We compare the accuracies and provide a method for looking at other predictive metrics and what they mean when evaluating a model by building the equivalent GUI.*

## **Purpose**

*The purpose of this study is to create a classification model that classifies a person's spine as healthy or unhealthy based on the 12 parameters present in the lower back pain symptoms dataset. The models are created using Machine Learning algorithms. The performance analysis is done to show which algorithm yielded a better accuracy in classification. Apart from creating a classification model, a GUI model was created to give more insight into differentiating between an unhealthy and healthy spine.*

## **Document Conventions**

*The whole SRS is written by IEEE 830 convention specifically (Copyright © 1999 by Karl E. Wiegers). The Entire project will be worked on GITHUB repository (*[*https://github.com/kushjayankpandya/SEM-6-Minor-Project*](https://github.com/kushjayankpandya/SEM-6-Minor-Project)*). The whole project is divided into three parts. The first is the preprocessing of Data. Second, finding the best model which suits the dataset. And third Being deploying the model with a GUI implementation as an application. The DFD and UML Use Case Diagrams are built for the last part (i.e Deploying the model).*

## **Intended Audience and Reading Suggestions**

*This project is being made as a minor project by KIIT students. This is a Machine Learning Project. We have collected the Dataset, preprocessed it. And on that data, We found out the best model for it and Deployed it.*

## **Product Scope**

*If any abnormality in a person’s spine is detected at an early stage then the patient can be treated to completely eliminate the problem before it leads to further complications. With the help of Machine Learning, we can create technology that can detect these abnormalities at an early stage.*

## **References**

*[1]. Minne H, Leidig G, Wuster C, Siromachkostov L, Baldauf G, Bickel R, Sauer P, Lojen M, R Ziegler .: A newly developed spine deformity index (SDI) to quantitate vertebral crush fractures in patients with osteoporosis. Bone and Mineral 3 335–349 1988 & Maturitas 10(3):248, 1988*

# **Overall Description**

## **Product Perspective**

# *The tool used for performing analysis on the data is Python. The aim was to create a classification model with higher accuracy than existing research done. Out of 310 observations, 67% of the data was used for training and 33% of the data was used for testing. The sample function was run on the entire dataset to generate sets of training and testing data. This was done to get a performance score of each model generated using all of the classification methods used in this research.*

## **Product Functions**

*This product shows us whether the spine details you added are normal or abnormal. You need to input the value such as pelvic\_incidence, pelvic tilt etc into the Application. And submit it and the App gives out results.*

## Technology Used

Programming Language - Python

Ide - Jupyter Notebook, Google collab

Third-party libraries - pandas, NumPy, matplotlib, TensorFlow,pyinstaller, Tkinter

Technology type - Machine Learning, Deep Learning, GUI

## **Operating Environment**

*This application can run on any Desktop PC. There is no internet connection required for its main task. But for the help button, it will be redirected to a webpage.*

## **Design and Implementation Constraints**

*The most important Implementation Constraint is the Dataset on which we would be training our ML models. If the models are not up to the required accuracy. It will be a big problem. Also, the dataset should be mostly complete. Some data missing in an example might lead us to not consider it while training, leading to a smaller dataset.*

## **User Documentation**

*A small help page showing how our software works and what it does will be put up on GITHUB. It will describe the project Medical Importance and its working.*

## **Assumptions and Dependencies**

*The whole project is done on python by using many third-party libraries. So Design and implementation are very much dependent on them. For example, TensorFlow, a Deep Learning Library can perform very well with GPU support. And its Installation is a hard task to implement.*

*Also, a Third-party library called pyinstaller which converts the python code into an executable program is also an important dependency without this working properly we will not be able to convert our python code into an executable file.*

# **External Interface Requirements**

## **User Interfaces**

*The User Interface is kept as simple as possible. There would be two buttons (i.e Help and Submit). Along with this, there would be twelve entry boxes where users have to input different attributes labelled around them. ‘Submit’ button will open a different window displaying the result. And with the ‘help’ button you would be directed to a webpage. The whole of the UI will be made using a Tkinter.*

## **Hardware Interfaces**

*The hardware can run on a PC environment supported by Pyinstaller.*

# **System Features**

## Highly Accurate ML model

**4.1.1 Description and Priority**

*The Application makes sense only if we are correctly able to predict the outcome. It’s a High priority feature with the relative risk being 5.*

**4.1.2 Functional Requirements**

*To find this model we would have to iterate over many approaches using a caviar approach to find the best model fit. We use Scikit-learn and TensorFlow libraries to do this. It also should be feed data which is clean and consistent which is done by the Pandas library.*

## A Decent GUI

**4.2.1 Description and Priority**

*A simple and fast GUI. It’s a High priority feature with the relative risk being 1.*

**4.2.2 Functional Requirements**

*To achieve this we would use Tkinter. And does not have any more functional Requirement.*

# **Other Nonfunctional Requirements**

## Help Page

*A simple help webpage will be uploaded on Github which can be utilized by the user in case of complications.*

# Future Enhancements

*Since we are dealing with a Medical Problem, it will be more suitable for the end-user to have easier access to this. So, we can deploy the model on an Android App and with the help of an API.*

**Appendix: Analysis Models**

UML - Use Case Diagram





