Cairo University
Faculty of Computers and Artificial Intelligence
Department of Computing and Bioinformatics

Diagnosis of Spinal Cord Injury

SCI
Spinal Cord Injury

Supervised by

Dr. Ahmed Farouk

Dr. Sabah Sayed

TA. Sarah Ahmed

Implemented by

Habiba Mustafa Ibrahim 20198024
Maryam Ali Adam 20198080
Maryam Sayed Ghanem 20198079
Omar Khaled Mohamed 20198124
Rania Mohamed Ghanem 20198033

Graduation Project
Academic Year 2022-2023
Final Documentation

Acknowledgement

First and foremost, thanks to **ALLAH** for helping us complete our graduation project.

Our deepest thanks and gratitude to Dr. **Ahmed Farouk** and Dr. **Sabah Sayed** for their constant direction and guidance.

Special thanks to Eng. **Sarah Ahmed** for the tremendous effort.

Abstract

Spinal cord injuries (SCI) are very common and may cause patients to become paralyzed or stop organs from functioning properly. Early detection of SCI is extremely useful which is why there are several approaches for SCI detection. However, these approaches do not make use of the power of artificial intelligence and thus face many challenges. Machine learning represents a promising frontier in epidemiological research on spine surgery. It consists of a series of algorithms that determines relationships between gene expression data and spinal cord injury. Analyses using machine learning algorithms have been applied on both traumatic SCI and nontraumatic SCI to help in the detection of the genetic biomarkers and the differences in gene expression between both traumatic SCI and nontraumatic SCI to help early prediction of Spinal cord injury. Evidence has shown that SCI is correlated with lots of genetic factors. That is why this project aims at conducting research in which we build and evaluate several machine learning models to detect non-traumatic SCI from human gene expression data. Various machine learning algorithms such as logistic regression, support vector machine, naive Bayes, and so on are used and their results are reported. Moreover, a web application is built deploying the most accurate models so that they can be used by neurologists providing a powerful tool for SCI diagnosis and gene analysis. The results showed that the highest accuracy is achieved using logistic regression with feature selection and data augmentation, with an accuracy of 85.71%. This indicates that the selected features and the augmented data helped to increase the accuracy of the model and improve its performance for the classification task.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Acronym	Full Term
ALS	Amyotrophic Lateral Sclerosis
CNS	Central Nervous System
CSS	Cascading Style Sheets
CSF	Cerebral Spinal Cord Fluid
DT	Decision Tree
ERD	Entity Relationship Diagram
GEO	Gene Expression Omnibus
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HTML	Hyper Text Markup Language
JS	JavaScript
KNN	k-Nearest Neighbors
MRI	Magnetic Resonance Imaging
LR	Logistic Regression
ML	Machine Learning
NB	Naive Bayes

NN	Neural Networks
RNA	Ribonucleic Acid
SC	Spinal Cord Injury
SCI	Spinal Cord Injury
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine

Chapter 1
Introduction

1.1. Background

The Spinal Cord (SC) is a major pathway for motor and sen-sory signals traveling between the brain and the peripheral-nervous system. The SC, along with the brain, comprises the central nervous system. It is tubular in shape and contains white matter (spinal tracts) and gray matter (neuronal-cell bodies). When a Spinal Cord Injury (SCI) occurs, the spinal tracts which convey sensory, motor, and autonomic-signals between the brain and organs are disrupted. An SCI may cause patients to become paralyzed or stop organs from functioning properly.

Figure 1.1 normal spinal cord

SCI is a serious medical condition that can result from trauma, disease, or degenerative disorders. It is estimated that there are approximately 17,000 new cases of SCI in the United States each year. SCI can cause a range of physical and neurological symptoms, including paralysis, loss of sensation, and bowel and bladder dysfunction. It can also have

significant psychological and social impacts on individuals and their families. The causes of SCI can vary widely, but the most common cause is trauma, such as a motor vehicle accident or fall. Other causes include medical conditions such as tumors or infections, and degenerative disorders such as multiple sclerosis or amyotrophic lateral sclerosis (ALS). Certain populations, such as athletes and military personnel, may also be at higher risk for SCI due to the nature of their activities.

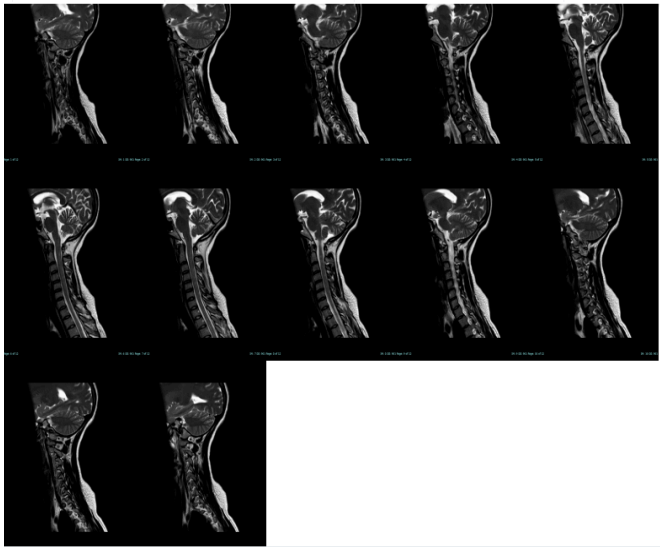


Figure 1.2 Non-Traumatic spinal cord

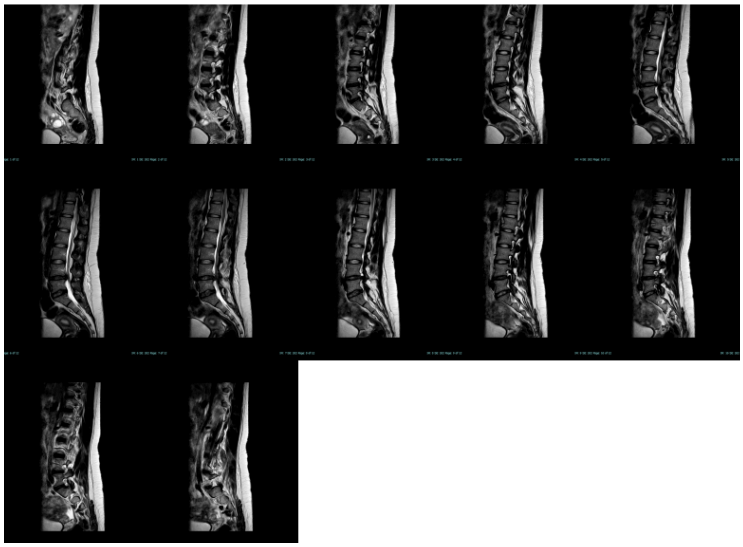


Figure 1.3 Traumatic spinal cord

Several cases highlight the devastating impact that spinal cord injuries can have on individuals' lives. For instance, consider a young athlete who sustains a spinal cord injury during a sports event. This sudden and traumatic event can result in permanent paralysis, forcing them to adapt to a completely different way of life. The individual may require extensive rehabilitation and assistive devices to regain some independence. Similarly, car accidents can also lead to severe spinal cord injuries, leaving victims with long-term disabilities and requiring ongoing medical care. These cases demonstrate the urgent need for improved methods of predicting and preventing spinal cord injuries, as they can have profound physical, emotional, and financial implications for those affected.

A search for genes that promote or block central nervous system (CNS) regeneration requires numerous approaches. For example, tests can be made on individual candidate molecules. Here, however, we describe methods for comprehensive identification of genes up- and down-regulated in neurons that can and cannot regenerate after injury. One problem concerns identification of low-abundance genes out of the 30,000 or so genes expressed by neurons. Another difficulty is knowing whether a single gene or multiple genes are necessary. When microchips and subtractive differential displays are used to identify genes turned on or off, the numbers are still too great to test which molecules are actually important for regeneration.

There are several approaches for SCI detection such as Magnetic Resonance Imaging (MRI) which provides invaluable information on severity and spinal cord level of injury. However, it is not always available and may be contraindicated for certain patients (e.g., those with injuries with penetrating metal). In addition, SCI often investigates protein changes in cerebral spinal cord fluid (CSF), but this is very challenging.

Therefore, machine learning (ML) can be used to facilitate the prediction of the outcomes in spinal cord injury on both traumatic SCI and non-traumatic SCI which was a challenging task before. It consists of a series of algorithms that determines relationships between genes and the presence of spinal cord injury and selecting the best features to work with to improve the accuracy of the model.

1.2. Motivation

Spinal cord injury is a devastating condition that can cause permanent disability and significantly impact a person's quality of life. Currently, there is no cure for SCI, and treatment options are limited to managing symptoms and preventing further damage to the spinal cord. Luckily, machine learning models have shown promise in predicting disease outcomes using gene expression data, and developing an app that can predict the likelihood of SCI could help identify at-risk individuals and potentially lead to earlier interventions.

When SCI occurs, symptoms depend on the severity of injury and its location on the spinal cord. Symptoms may include partial or complete loss of sensory function or motor control of arms, legs and/or body. The most severe spinal cord injury affects the systems that regulate bowel or bladder control, breathing, heart rate and blood pressure. Most people with spinal cord injury experience chronic pain. An SCI may cause patients to become paralyzed or stop organs from functioning properly. So, this motivated us to work on this to help people detect it early and try to avoid these issues.

Understanding genetic factors and genetic disorders is important in learning more about promoting health and preventing disease, and spinal cord injury is a type of damage to the nerve tract that runs from the lower back to the brain. As a high-cost neurological disability, SCI can even lead to permanent paralysis and loss of sensation. Evidence has shown that SCI is correlated with lots of genetic factors and that’s why we will use this study to detect the genes that are responsible for the SCI.

Machine learning models have shown promise in predicting disease outcomes using gene expression data, and applying these models to SCI could have significant implications for the field. By incorporating data from large-scale genomic studies and clinical trials, an app could provide personalized risk assessments and recommendations for preventative measures and interventions. In addition, it could help researchers identify novel targets for therapy and accelerate the development of new treatments for SCI.

Moreover, developing an app that can predict the likelihood of SCI could help identify individuals who are at risk and provide them with information and resources to prevent or manage the condition. It could also support research efforts to better understand the underlying mechanisms of SCI and develop more effective treatments.

1.3. Problem Definition

Diagnosis of spinal cord injury severity at the ultra-acute stage is of great importance for emergency clinical care of patients as well as for potential enrollment into clinical trials. However, the lack of a diagnostic biomarker for SCI has played a major role in the poor results of clinical trials. Moreover, there are not any tools that help neurologists understand genetic factors and genetic disorders related to SCI. Therefore, a method for detecting SCI from the patients' genetic data is required along with an analysis of these genes.

Despite advances in medical technology and rehabilitation, there is currently no cure for SCI. Treatment options are limited to managing symptoms and preventing further damage to the spinal cord. Research into new therapies and interventions is ongoing, but progress has been slow due to the complexity of the condition and the challenges of conducting clinical trials.

1.4. Project Objective

The objective of this project is to conduct research in which we build and evaluate several machine learning models to perform early detection of non-traumatic SCI from gene expression data as well as develop a prognostic tool (web application) that can predict spinal cord injury through identifying and analyzing the genes that affect it.

Since it is one of the most promising solutions for predicting SCI, we are using machine learning models and developing an app that can analyze an individual's gene expression data. The app could incorporate data from large-scale genomic studies and clinical trials to identify biomarkers and genetic variants associated with SCI. By training machine learning algorithms to identify patterns and correlations in this data, the app could develop predictive models for early diagnosis and intervention.

The proposed application would be designed to take in an individual's genetic data and provide a personalized risk assessment for SCI, based on their genetic profile, to reduce the risk of SCI. Additionally, it analyzes the patients' data enabling the user to stay informed and engaged in their patients care. The proposed application uses gene expression to detect if the person's gene related to the SC are being over expressed which

leads to traumatic SCI or if the genes are being under expressed which can leads to recovery if he already suffered from SCI or if the genes are are being regularly expressed which means that there is no further damage to the human body. The application can be used by a neurologist to find out if his patient is healthy, with a non-traumatic injury or if he has spinal cord injury just by uploading the patient gene expression file.

This study uses different machine learning algorithms for SCI diagnosis such as logistic regression (LR), naive Bayes (NB), neural networks (NN) and more. By incorporating machine learning models into the app, it could continuously learn and improve its predictive capabilities over time. This could lead to more accurate and personalized risk assessments for SCI, as well as more effective interventions and treatments. Ultimately, a predictive app for SCI could have significant implications for the field, enabling earlier diagnosis and intervention, improving outcomes for individuals with SCI, and supporting ongoing research efforts to better understand and treat this condition.

1.5. Gantt Chart Of Project Time Plan

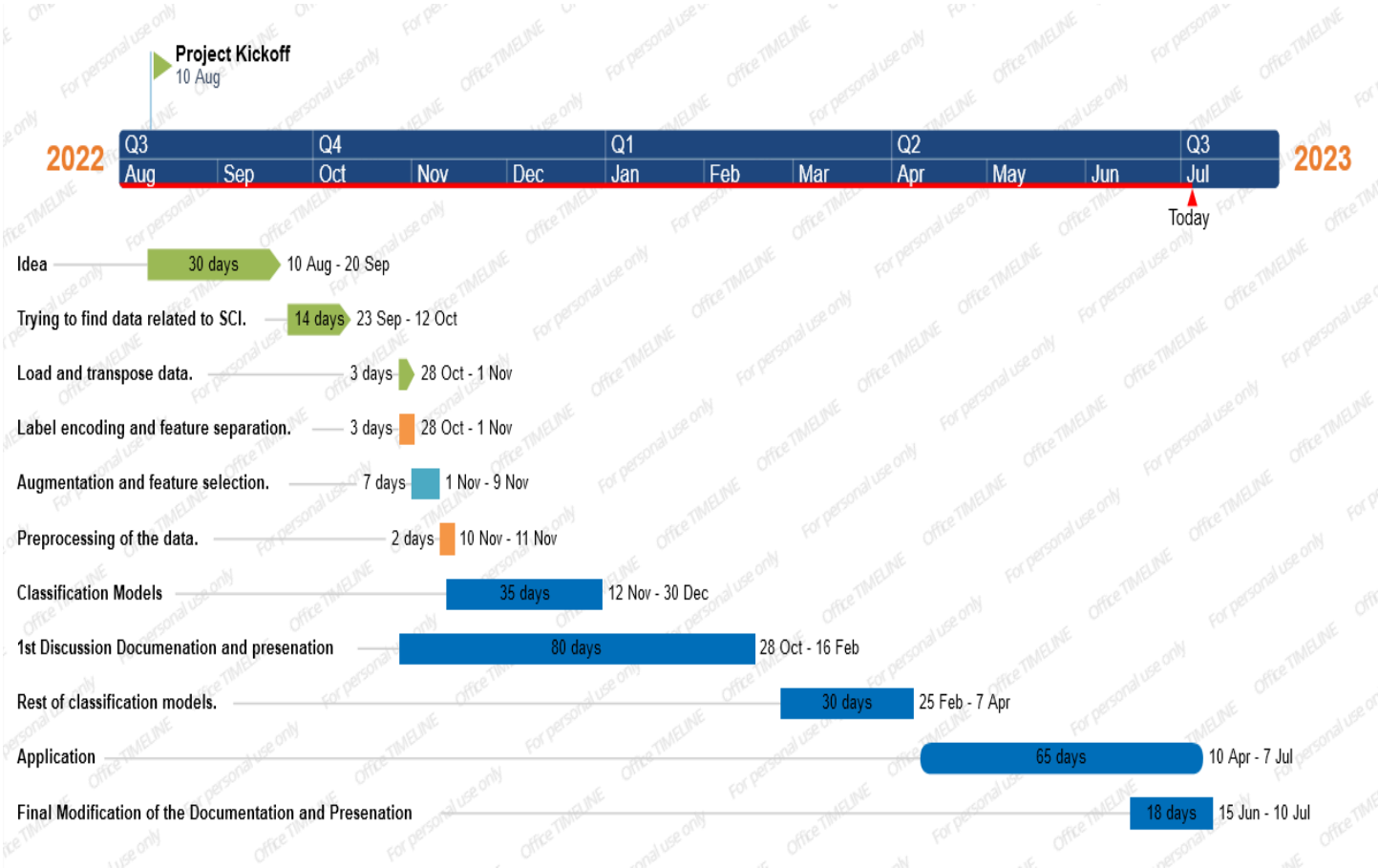


Figure 1.4 Gantt chart

1.6. Project Development Methodology

We used the "Waterfall" methodology in developing the web application and "Agile" methodology in the model building process in this project that uses machine learning for classifying spinal cord injury. This can help ensure the project is well-planned, efficiently

executed, and successful in meeting the project objective .

Here are the details of how we used the Waterfall and Agile methodologies in this project:

- **Planning phase (Waterfall):** In the planning phase, the project scope and timeline are defined upfront, and any changes to these parameters can be difficult to accommodate. This phase is best suited for well-defined tasks that have clear outcomes, such as building the website. We used the Waterfall approach to define the website's features, design, and functionality, and create a detailed project plan that outlines the tasks, timelines, and deliverables.
- **Research and data preparation phase (Agile):** In the research and data preparation phase, the team used an Agile approach to explore the gene expression data and identify the most relevant features for classifying spinal cord injury. The team worked in short sprints, using iterative cycles to refine the data preparation process, select the appropriate machine learning model, and optimize the model's hyperparameters to achieve the best performance.
- **Model training and testing phase (Agile):** In the model training and testing phase, the team used Agile to train and test the machine learning model, fine-tune the parameters, and evaluate the results. The team used an iterative approach to refine the model, test different algorithms, and explore different feature selection techniques to improve the model's accuracy and efficiency.
- **Website implementation phase (Waterfall):** In the website implementation phase, we used a Waterfall approach to implement the website design, develop the HTML code, and integrate Flask to build the website's functionality. We worked through the project plan, completed each task in sequence, and ensured the website meets the requirements set out in the project plan.

By using a combination of Waterfall and Agile methodologies, we were able to ensure that the project is well-planned, efficiently executed, and successful in meeting the requirements.

1.7. The Tools Used in the Project

The project utilizes a variety of software and hardware tools to develop the

application that can predict spinal cord injury using gene expression data. These tools will be discussed in the following paragraphs.

In terms of software, the project uses programming languages such as Python and Flask for handling the machine learning models and server-side scripting. Python is a popular language for machine learning due to its extensive libraries and frameworks while Flask is commonly used for server-side web development. Additionally, the project makes use of web technologies such as HTML, CSS, and JavaScript for building the user interface and creating a user-friendly experience.

Machine learning libraries such as scikit-learn and TensorFlow are also utilized in the project for developing the machine learning models that predict spinal cord injury. These libraries offer pre-built models and algorithms that can be customized and trained on the dataset used in the project.

In terms of hardware, the project requires a computer with sufficient processing power to handle the machine learning algorithms and process large datasets. Additionally, the project may require specialized hardware such as a graphics processing unit (GPU) for faster model training and inference.

1.8. Report Organization

In chapter 1, an introduction was given in which we talked about the motivation that drove us to do this idea for our project, the definition of our problem and why it is a very important and critical topic, the objective of our project and our solutions. Also, a Gantt chart of the project time plan is shown and our development methodology and used tools are explained.

In chapter 2, we talk about the related work of the project and the differences between them and our project and how our project is useful even with this related work.

Chapter 3 presents the project specification including the functional requirements and nonfunctional requirements and the use case diagram of the project.

In chapter 4, the design is presented with multiple figures of the project diagrams including system component diagram, system class diagram, sequence diagram, project entity relationship diagram (ERD) and graphical user interface (GUI) design.

Chapter 5 discusses the implementation and testing of the application and includes detailed information about the materials and methods like dataset and machine learning models and how they were implemented.

In chapter 6, we show and discuss the results obtained from our project. Finally, chapter 7 gives the concluding remarks of this project.

Chapter 2
Related Works

Studies performed on the detection of spinal cord injury:

- **Spinal cord injury in rats [1]:**

  Most types of spinal cord injuries seen in humans can be replicated in adult rats. These include complete and incomplete spinal cord injuries at different levels. The epicenter area of the injured spinal cord was isolated for RNA extraction and processing and hybridization on Affymetrix GeneChip arrays. In order to understand the distinct molecular events underlying this injury model, an analysis of global gene expression of the acute, subacute and chronic stages of a moderate to severe injury to the rat spinal cord was conducted using a microarray gene chip approach. Some related genes were identified using bioinformatics analysis of gene expression.

- **Spinal cord injury in monkeys [2]:**

  Small cysts formed in the monkey spinal cord resulted from tissue necrosis, posttraumatic axonal degeneration, and demyelination, particularly following the 1.5 mm tissue displacement injury. The aim of this study was to verify the effective use of PNG and aFGF for repairing incomplete SCI in nonhuman primates. Postoperative behavioral evaluations, electrophysiologic tests (including motor and somatosensory evoked potentials), and magnetic resonance imaging were performed and compared between the 2 groups as well as histologic examination of the spinal cord cephalic to, at, and caudal to the lesion site after sacrifice. A combined electrophysiological, MR imaging, and histology assessments allowed to detect severity-dependent changes in the monkey following injuries produced.

- **Spinal cord injury in humans:**

  In one study **[3],** the authors detect if humans are healthy individuals without history of CNS pathology or non-CNS trauma patients or individuals who suffered a traumatic spinal cord injury. They analyzed global gene expression in peripheral white blood cells during the acute injury phase and identified 197 genes whose expression changed after SCI compared to healthy and trauma

controls and in direct relation to SCI severity. Unsupervised co-expression network analysis identified several gene modules that predicted injury severity (AIS grades) with an overall accuracy of 72.7%.

Also, in another study on humans **[4],** the authors stated that they reached images in kaggle but they were not labeled and, so the aim was to detect the region of interest in spinal cord injury instead of SCI classification.

One such study **[5]** published in the journal Scientific Reports utilized a machine learning approach to identify gene expression patterns associated with SCI. The study used microarray data from 60 SCI patients and 19 healthy controls to identify differentially expressed genes and develop a machine learning model for predicting SCI. The model achieved an accuracy of 84.2% in predicting SCI, demonstrating the potential of using gene expression data and machine learning models for SCI diagnosis.

Another related work **[6]** was published in the journal BMC Bioinformatics, which utilized a machine learning approach to predict the outcome of SCI based on gene expression data. The study used microarray data from 102 SCI patients and 15 healthy controls to identify differentially expressed genes and develop a machine learning model for predicting SCI outcomes. The model achieved an accuracy of 78.3% in predicting SCI outcomes, demonstrating the potential of using machine learning models for predicting disease outcomes and developing personalized interventions.

The main differences between those studies and our project are:

While there are similarities between this project and the related works discussed earlier, there are also some differences. This project aims to develop an application that can predict spinal cord injury (SCI) using gene expression data. The machine learning algorithms used in this project include Logistic Regression, Naive Bayes, Neural Networks, and decision tree. The app is built using web technologies such as HTML, Flask, and Python.

In contrast, the related works discussed earlier focus on using machine learning models to diagnose SCI using gene expression data. Those studies identify differentially expressed genes and develop machine learning models to

diagnose SCI and its outcomes. The models use techniques such as RNA sequencing and microarray data analysis to identify biomarkers and genetic variants associated with SCI.

Therefore, the key difference between this project and the related works is the focus of the research. This project aims to develop a predictive model for SCI, while the related works aim to develop diagnostic tools for SCI and its outcomes. Additionally, this project uses a combination of machine learning algorithms to analyze the gene expression data, while the related works use a variety of data analysis techniques such as RNA sequencing and microarray data analysis.

Also, this study is working with human data, specifically on the gene level. A variety of machine learning algorithms have not yet been applied to the dataset we are working with, and that offers a chance for improved results and more accurate SCI detection. This project aims to leverage machine learning models and gene expression data to improve outcomes for individuals with SCI.

Chapter 3
System Analysis

3.1. Project Specifications

3.1.1. Functional Requirements

ID	1
Function	Sign up
Description	This function allows the user to create account
Action	The user is added into the database
Requirements	Performed by the user
Input	Username - Email - Password - Confirm password
Output	Returns account created or an error creating
Precondition	Using username and email that hasn't been registered into the database
Postcondition	Users added into database

ID	2
Function	Sign In
Description	This function allows the user to access their account
Action	The user's login credentials are verified against the database
Requirements	Performed by the user
Input	Username and password
Output	Signing user into his account or showing invalid credentials error message
Precondition	Valid Email and password
Postcondition	User is logged into their account

ID	3
Function	Log out
Description	This function allows the user to log out of his account
Action	The user's session is terminated and they are logged out of their
Requirements	Performed by the user
Input	N/A
Output	Logging the user out of his account and redirect him to the home page
Precondition	User is logged into their account
Postcondition	User is logged out of their account and their session is terminated.

ID	4
Function	Upload Gene Expression File
Description	This function allows the user to upload gene expression file to the system
Action	The gene expression file is uploaded and stored in the database
Requirements	Performed by the user
Input	Gene expression excel file
Output	Returns successful upload or an error uploading
Precondition	- It requires the user to be logged in into his account. - Uploaded file has to be excel file - Uploaded file has to contains at least 6290 genes related to SCI - Uploaded file has to contain only two column/rows one contains genes names and the other contains genes expression values
Postcondition	Gene expression files are uploaded and stored in the system

ID	5
Function	Get Spinal Cord Injury Detection Results
Description	This function allows the user to get the results of spinal cord injury detection using classification models built with machine learning
Action	The system displays to the user the file prediction and stores the prediction into the database
Requirements	Performed by the user
Input	N/A
Output	Returns the file prediction or an error message
Precondition	- It requires the user to be logged in into his account. - Uploaded file has to be excel file - Uploaded file has to contains at least 6290 genes related to SCI - Uploaded file has to contain only two column/rows one contains genes names and the other contains genes expression values
Postcondition	Prediction is stored into the database

ID	6
Function	Save File into Database
Description	This function allows the user to save a multiple files into the database
Action	The user's file is saved into the database once user uploads a file
Requirements	Performed by the user
Input	File to save
Output	Returns file added to your files history or an error message
Precondition	- It requires the user to be logged in into his account. - Uploaded file has to be excel file - Uploaded file has to contains at least 6290 genes related to SCI - Uploaded file has to contain only two column/rows one contains genes names and the other contains genes expression values
Postcondition	File is saved into the database

ID	7
Function	View All Files
Description	This function allows the user to view all files saved in the database
Action	Pressing on go to files history button display on user's account
Requirements	Performed by the user
Input	N/A
Output	Returns all files saved in the database or an error retrieving
Precondition	User is logged into their account
Postcondition	- It requires the user to be logged in into his account. - Uploaded file has to be excel file - Uploaded file has to contains at least 6290 genes related to SCI - Uploaded file has to contain only two column/rows one contains genes names and the other contains genes expression values

ID	8
Function	View Statistical Analysis Results
Description	This function allows the user to view the statistical analysis results applied on the gene expression files uploaded
Action	Pressing on show statistical analysis button
Requirements	Performed by the user
Input	N/A
Output	Returns the statistical analysis results
Precondition	It requires the user to have at least two files history of at least two files
Postcondition	Statistical analysis results are displayed to the user

ID	9
Function	Contact Us
Description	This function allows the user to get in touch with the system administrators by sending an email through the contact page
Action	User sending an email by pressing send message button displayed on the contact page
Requirements	Performed by the user
Input	User's name, email, subject, and message
Output	Returns Message sent successfully or an error message
Precondition	It requires the name field, Email field, subject field and message field to be all filled
Postcondition	User's email is stored into the system data

ID	10
Function	Access About Page
Description	This function allows the user to access the about page to learn more about Spinal Cord Injury (SCI)
Action	The system displays information about SCI on the about page to the user
Requirements	Pressing about button displayed on the app
Input	N/A
Output	Redirect the user to the about page
Precondition	N/A
Postcondition	Information about SCI is displayed to the user on the about page.

3.1.2. Non-functional Requirements

ID	1
Function	Usability
Description	This function ensures that the web app has a user-friendly interface and is easily used by any specialist

ID	2
Function	Response Time
Description	This function ensures that the system provides the user with fast actions and responses Example: user can get prediction on gene expression file within few seconds

ID	3
Function	Robustness
Description	This function ensures that the system can cope with invalid inputs Example: system reveals expressive error message in case the user uploaded empty files, files that contain genes that are not related to SCI or non gene expression files.

ID	4
Function	Security
Description	This function ensures that all data inside the system will be protected against unauthorized access Example: user's password is being hashed before being stored in the database, incase someone got access to the database he will not be able to view the actual password of the user

ID	5
Function	Reliability
Description	This function ensures that the system operates without failure in a specific environment and duration

3.2. Use Case Diagram

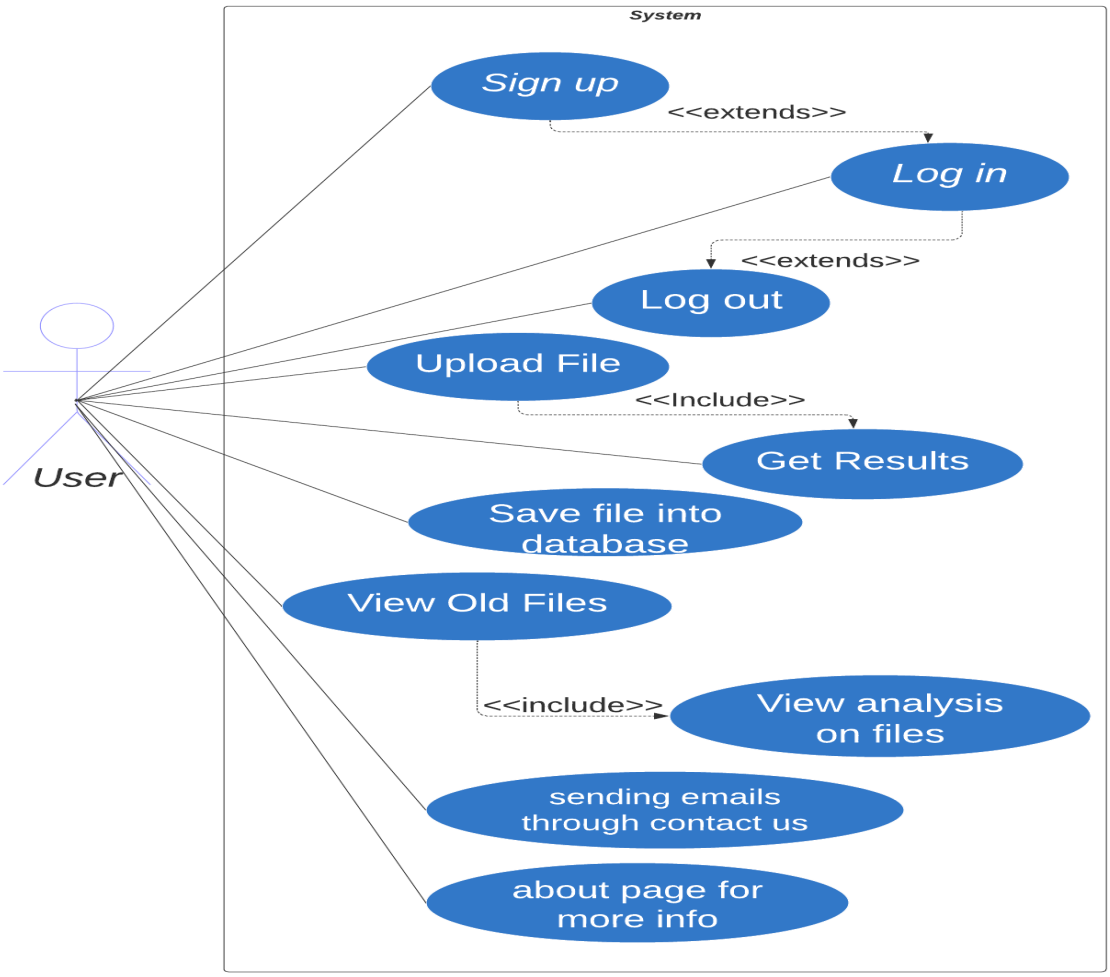


Figure 3.1: Use Case Diagram

Chapter 4
System Design

4.1. System Component Diagram

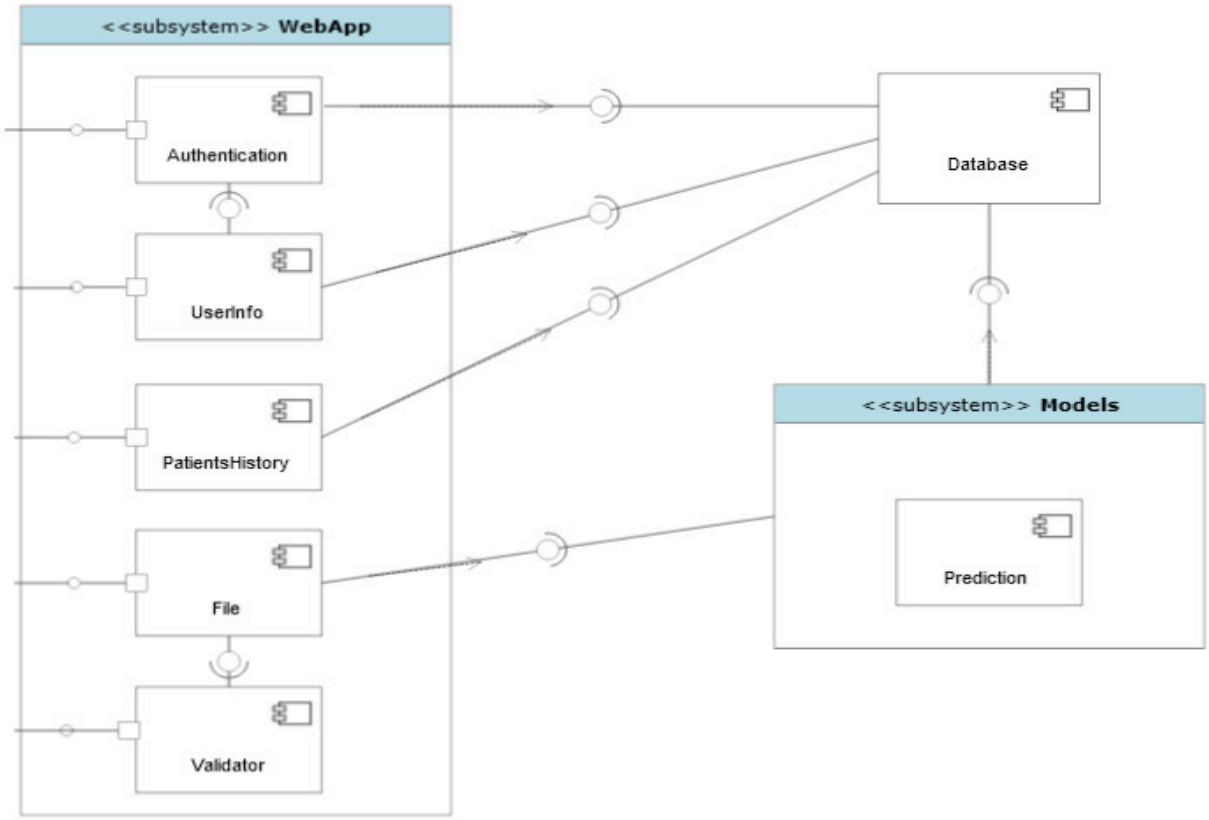


Figure 4.1: Component Diagram

4.2. System Class Diagram

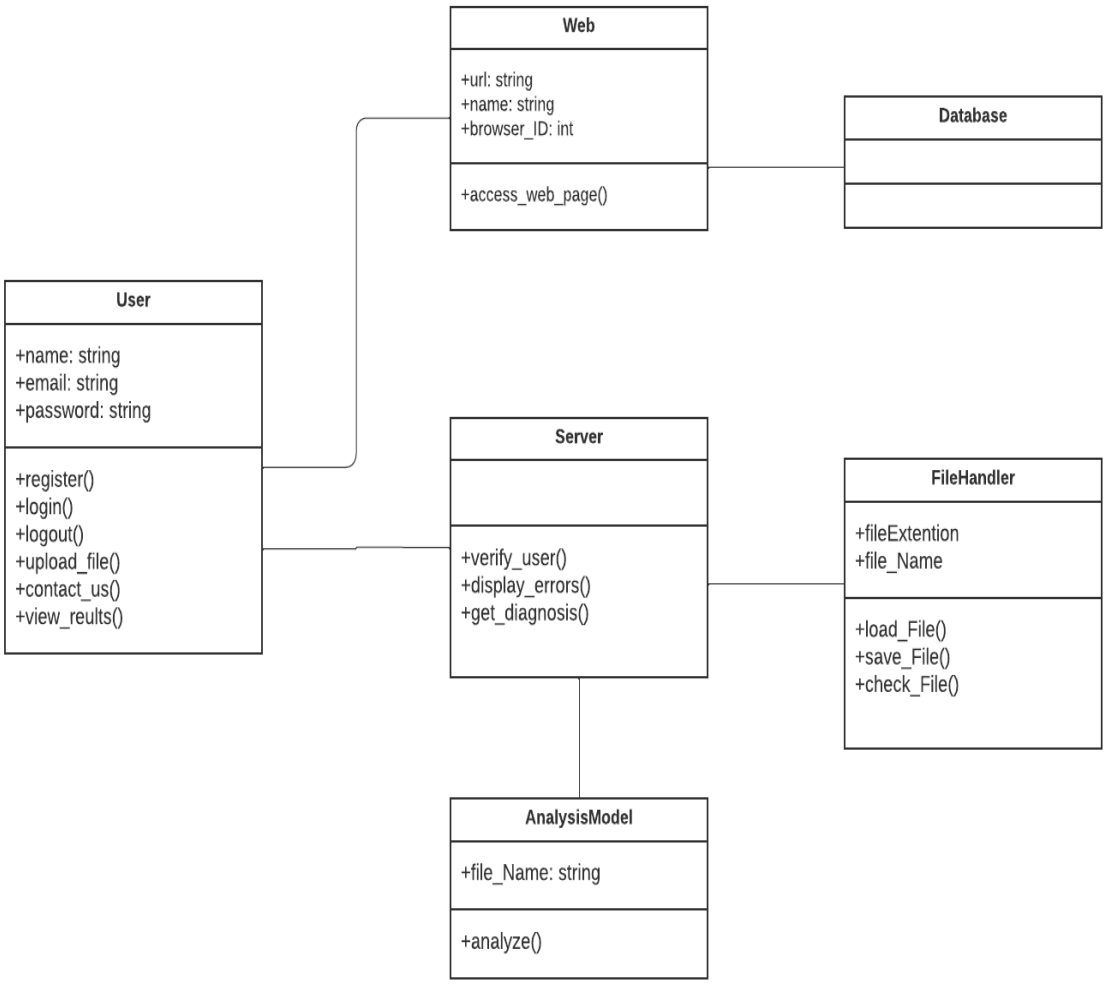


Figure 4.2: Class Diagram

4.3. Sequence Diagrams

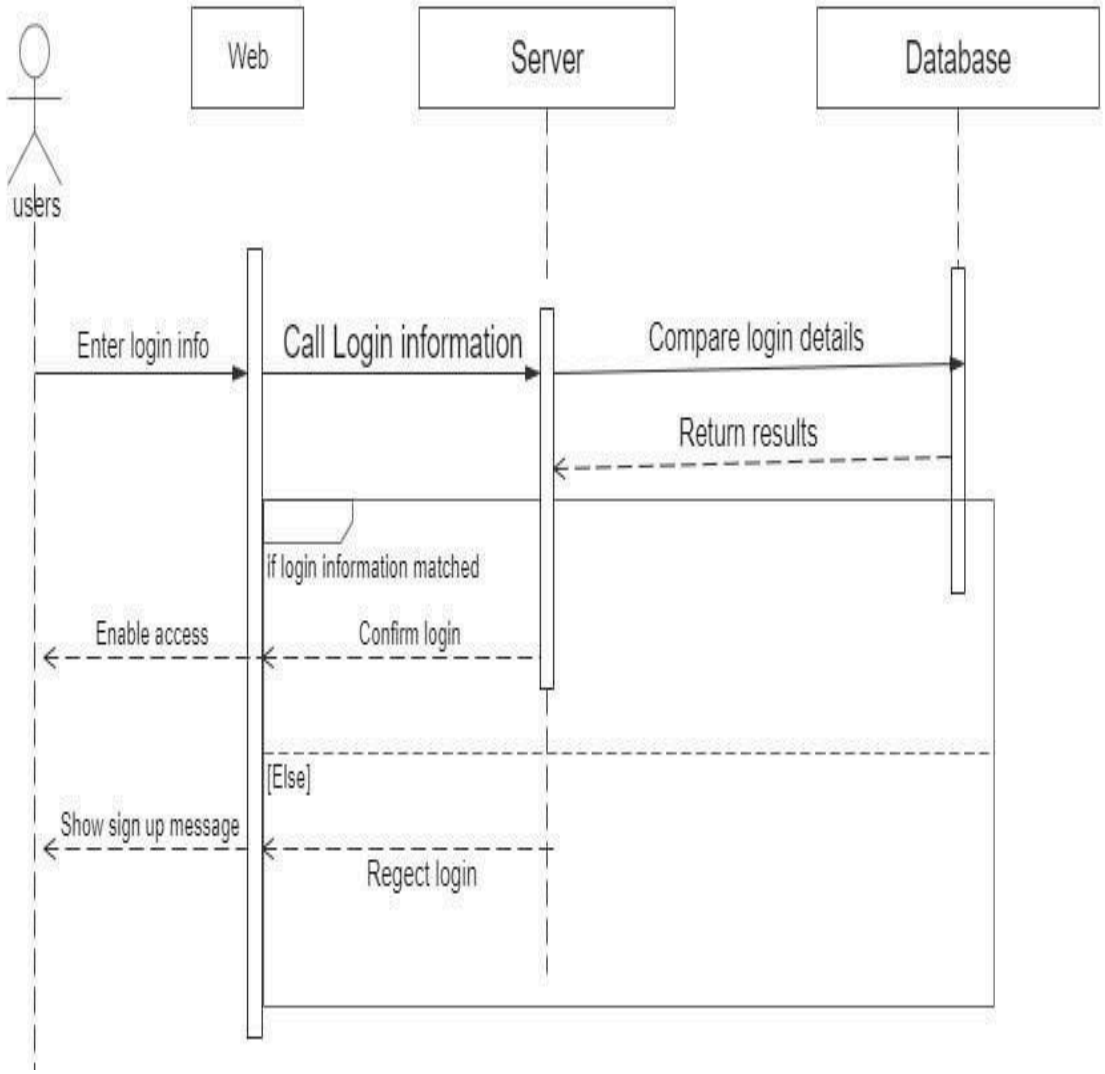


Figure 4.3: Login sequence diagram

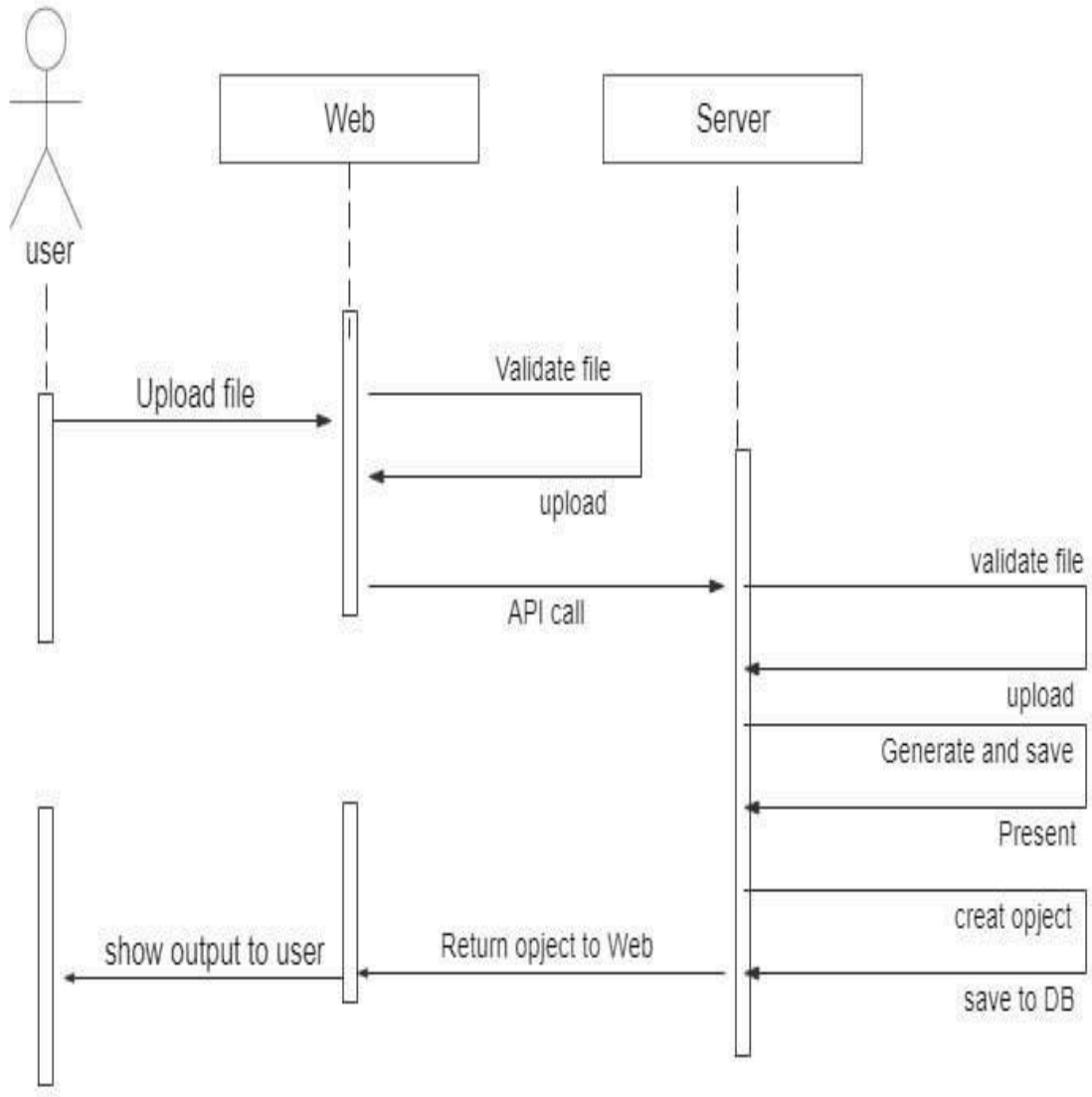


Figure 4.4: Upload File sequence diagram

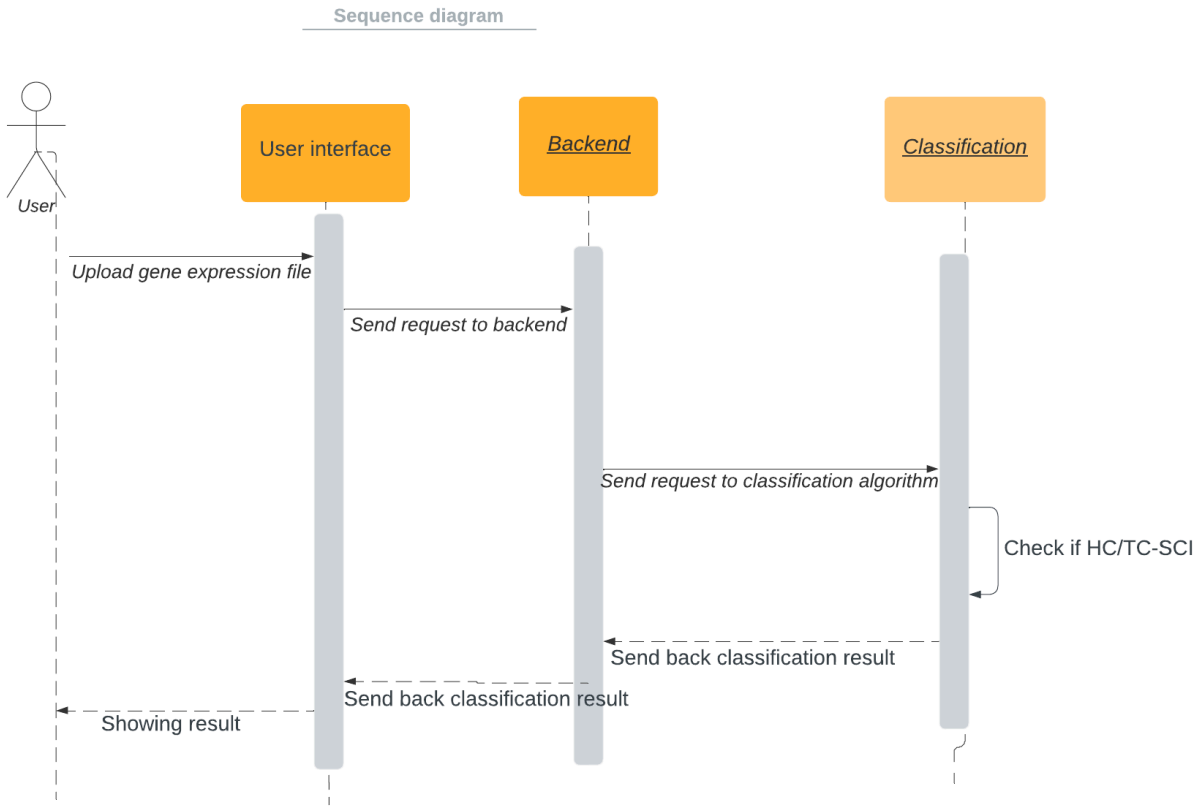
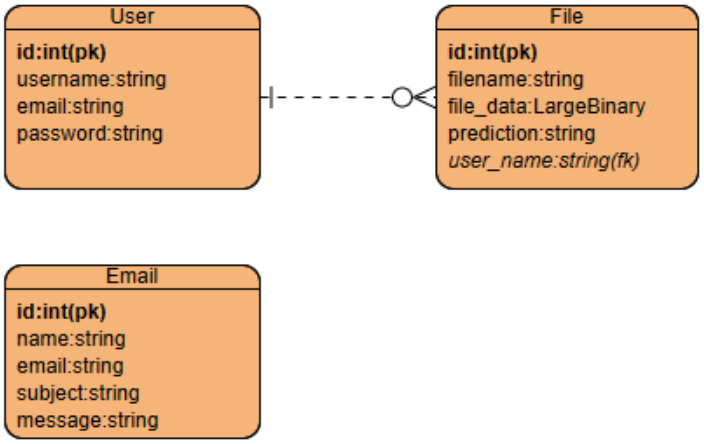


Figure 4.5: Get results sequence diagram

4.4. Project ERD



4.5. System GUI Design

Figure 4.6: Application Home Page

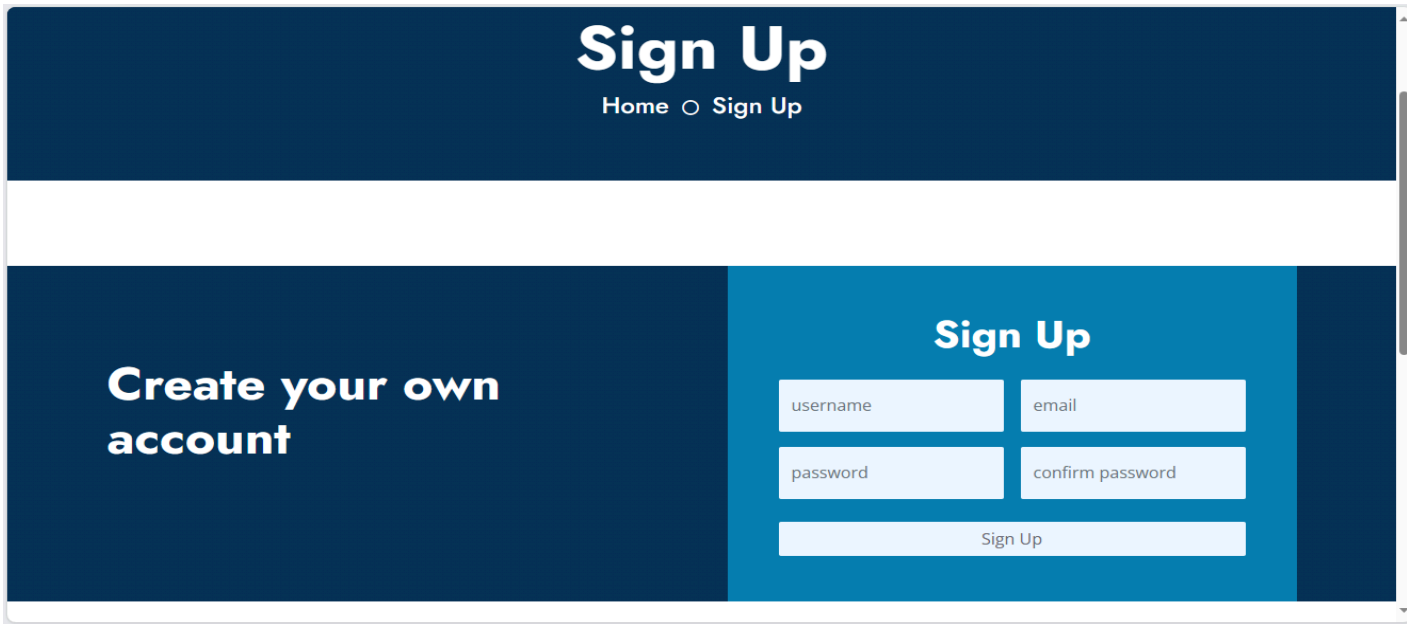


Figure 4.7: Sign Up Page

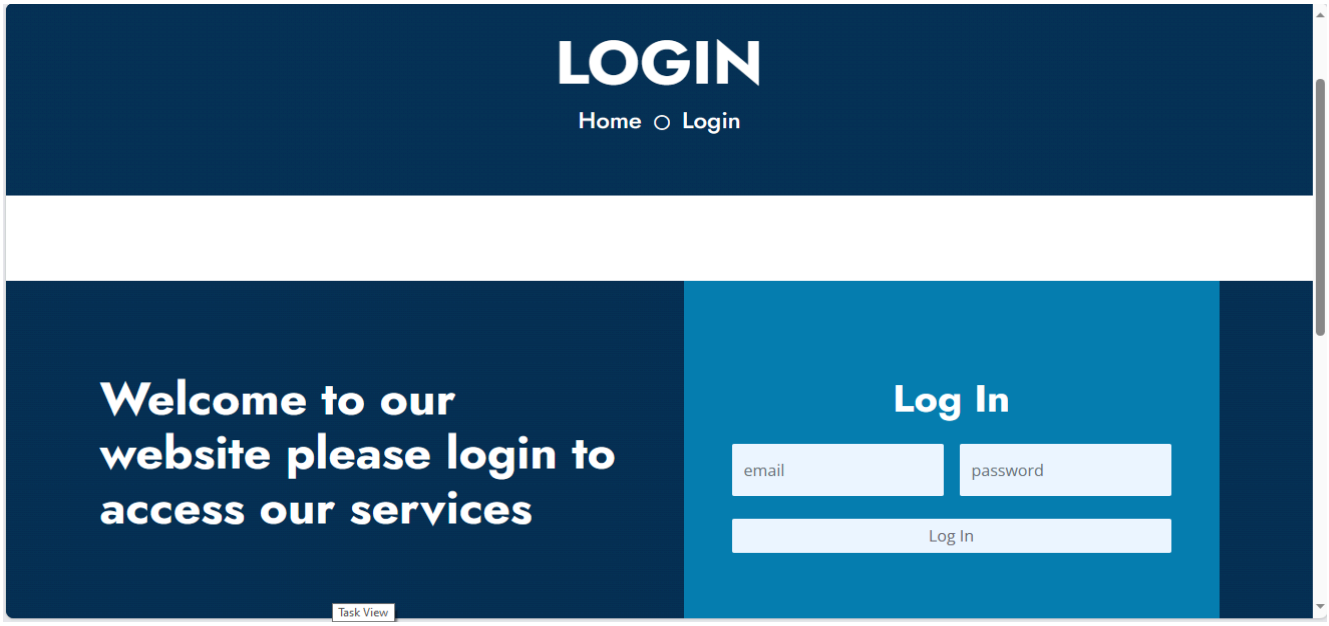


Figure 4.8: Login Page

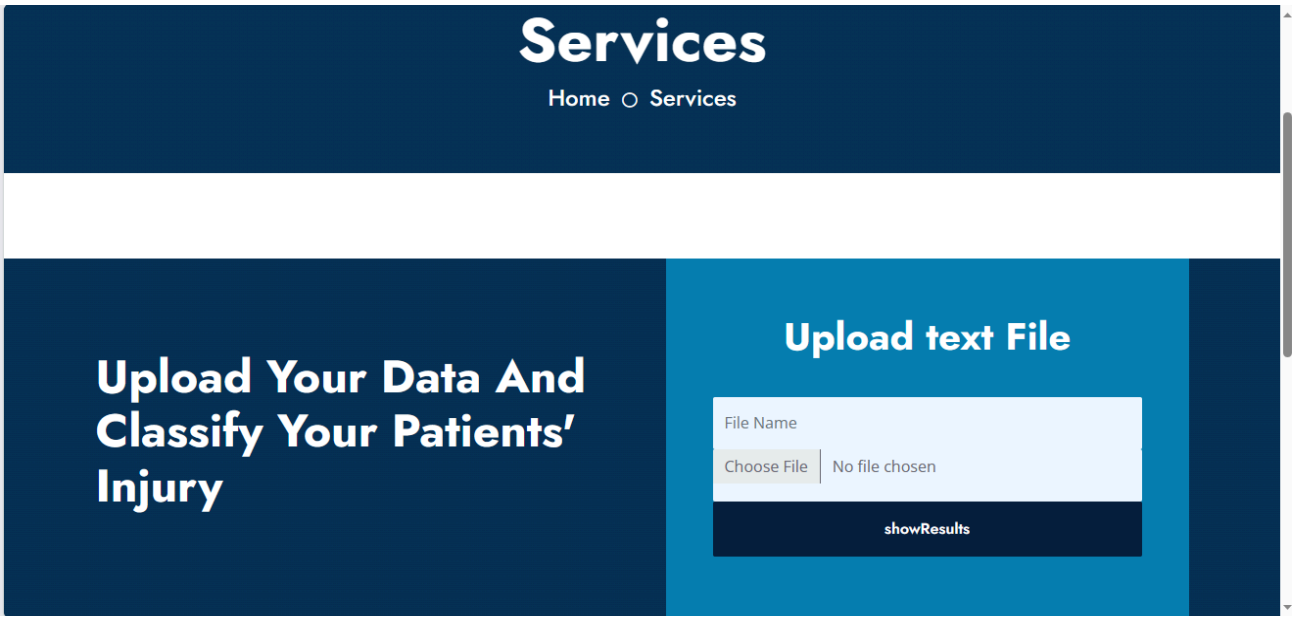


Figure 4.9: Upload File Page

Figure 4.10: About Us Page

Chapter 5
Materials And Methods

5.1. Dataset

The GSE151371 ribonucleic acid (RNA) -sequencing dataset, which contains blood RNA biomarkers for spinal cord injury, was downloaded from the Gene Expression Omnibus (GEO) database. The dataset consists of a normalized gene expression matrix for humans, comprising 17,501 genes and 58 sequenced samples.

Out of the 58 samples, 10 samples were obtained from healthy individuals with no history of central nervous system pathology, 10 samples were obtained from non-CNS trauma patients, and 38 samples were obtained from individuals who had suffered a traumatic spinal cord injury. This dataset provides a valuable resource for the identification of gene expression patterns associated with different types of SCI and the development of diagnostic and therapeutic tools for this debilitating condition.

	0	1	2	3	4	5	6	7	
0	Gene	HC01	HC02	HC03	HC04	HC05	HC06	HC07	H
1	LOC729737	6.17911108280235	6.58065796670347	6.8967625986374	5.40568524953635	6.71994616435615	5.91430085089696	5.77574687021244	5.20220451549
2	MIR6723	6.79101514004344	6.98802307165147	6.7420322106266	5.9910989351952	6.34226123864681	6.10193167835156	6.08978507704103	6.24669016049
3	LOC100133331	-3.45198803468764	-2.2564283915122	0.380932433132175	-3.91528382041089	-1.56075861148905	-2.45092426857319	-2.86316181296	-3.18741382878
4	LOC100288069	5.77198968587589	6.30262065959179	6.98969288037085	4.74586008274815	6.54238285313473	5.90866452825092	5.50778505424236	4.85332470404
...	...	...	...	...	...	...	...	...	
17496	KDM5D	-1.788215	6.621278	6.058275	3.861823	-3.980302	6.806247	6.765468	3.666
17497	TTTY10	-4.287254	0.537679	0.211321	-0.77314	-4.832722	0.749481	0.673642	0.329
17498	EIF1AY	-3.59682	4.036264	2.158366	2.895851	-5.081685	5.02329	4.307794	2.140
17499	PRORY	-1.114126	-0.850515	-0.574671	-2.500414	-1.885118	-2.588618	-2.662678	-0.775
17500	LOC101929148	-2.763978	-1.988858	-1.677034	-2.804171	-2.521827	-1.895274	-2.360151	-3.231

Figure 5.1: Normalized Gene Expression Matrix

The RNA-sequencing technology used in this study enables the measurement of gene expression levels by sequencing the RNA molecules present in the blood samples. The

resulting gene expression matrix contains quantitative information about the expression levels of thousands of genes, which can be used to identify differentially expressed genes between healthy individuals, non-CNS trauma patients, and SCI patients.

Overall, the GSE151371 dataset provides a valuable resource for us to develop diagnostic and therapeutic tools for spinal cord injury. By identifying biomarkers and gene expression patterns associated with different types of SCI, this dataset has the potential to improve outcomes for individuals with SCI and accelerate the development of targeted interventions

5.2. Pipeline

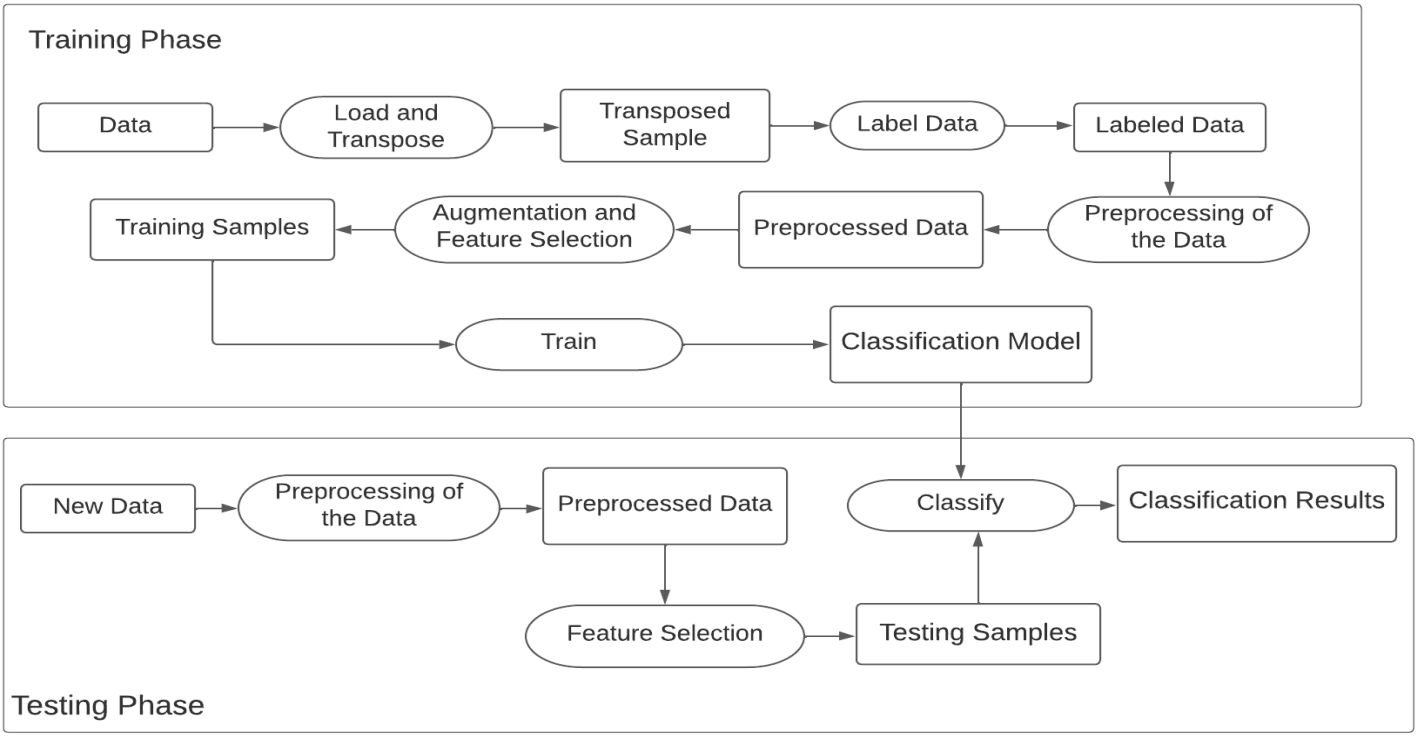


Figure 5.2 shows the pipeline of the model building process.

5.2.1. Data Collection

The classification of spinal cord injury involves a series of steps,

1. beginning with the collection of a suitable dataset. In this case, the dataset used was obtained from the Gene Expression Omnibus database and contains RNA sequencing data for 58 samples, including healthy individuals, non-CNS trauma patients, and SCI patients and 17501 different genes related to the SCI.
2.

5.2.2. Data Augmentation and Preprocessing

Once the dataset has been collected, the next step in machine learning is to load and preprocess the data. This involves a series of standard procedures that are necessary to prepare the data for use in a ML model. In addition to these standard procedures, data augmentation techniques such as oversampling and Synthetic Minority Over-sampling Technique (SMOTE) may also be used to increase the size of the dataset and improve the performance of the classification models.

Standard Preprocessing Techniques:

- **Normalization:** Normalization is a common preprocessing technique that involves scaling the data so that all features have a similar range of values. This can help to improve the stability and convergence of the ML models.

- **Shuffling:** Shuffling involves randomizing the order of the samples in the dataset. This can help to ensure that the model is not biased towards any specific order of the data, which can improve the generalization of the model.

- **Label Encoding:** Label encoding is a technique used to convert categorical labels into numerical values. This is necessary because many ML models require numerical inputs. Label encoding assigns a unique numerical value to each category in the label set. So, the encoding performed in this project was:

  Label "HC" was converted to 0

  Label "TC" was converted to 1

  Label "SCI" was converted to 2

Data Augmentation Techniques:

- **Oversampling:** Oversampling is a data augmentation technique that involves replicating samples from the minority class to balance the class distribution in the dataset. This can help to improve the performance of the models on imbalanced datasets.

- **SMOTE:** Synthetic Minority Over-sampling Technique is a data augmentation technique that involves generating synthetic samples from the minority class by interpolating between existing samples. This can help to increase the size of the dataset and improve the performance of the models on imbalanced datasets.

5.2.3. Feature Extraction

Feature extraction is an important step in machine learning that involves selecting a subset of features (or variables) from a larger set of input data that are most relevant to the problem being solved. In the case of diagnosing spinal cord injury using gene expression data, there are typically a very large number of genes that need to be considered, which can make the classification task computationally and statistically challenging.

To address this challenge, multiple techniques were used for feature extraction in the proposed project, including SelectFromModel, VarianceThreshold, and chi-square feature selection. These techniques help to identify the most informative genes that have a significant impact on the diagnosis of SCI, while reducing the dimensionality of the input data and improving the performance of the classification models.

SelectFromModel is a technique that uses a machine learning model to identify the most important features in the input data. The model is trained on the full set of input features, and then the most important features are selected based on a user-defined threshold. This technique can be computationally expensive, but it can be very effective at identifying the most informative features.

VarianceThreshold is a simpler technique that removes features with low variance. Features with low variance are less likely to be informative, as they do not vary

much across the input data. This technique can be computationally efficient and can help to reduce the dimensionality of the input data.

The chi-square feature selection method is a statistical technique that measures the dependence between two categorical variables. In the context of gene expression data, this technique can be used to identify the genes that are most strongly associated with the diagnosis of SCI, based on their expression levels in different samples.

After applying multiple experiments and testing these feature selection techniques with the gene expression dataset, it was found that chi-square produced the best results. Thus, it will be used in the project's ML pipeline.

5.2.4. Models Training and Testing

The final step in the classification process is building, testing, and evaluating different machine learning models. Several ML algorithms were used for this task, including logistic regression, support vector machine (SVM), decision tree (DT), naive Bayes, k-nearest neighbors (KNN), and neural networks. These algorithms are trained on the preprocessed and feature-extracted data and evaluated.

In the model training phase, the preprocessed and feature-extracted data is split into training and testing sets. The training set is used to train the machine learning model using a specific algorithm, such as logistic regression, SVM, or neural network. During the training process, the model learns the patterns in the data that are associated with SCI. The training process involves adjusting the model's parameters, such as weights and biases, to minimize the error between the predicted and actual classification labels.

Once the model is trained, the next step is to test its performance on the testing set. The testing set is used to evaluate how well the model can generalize to new, unseen data. The model's performance is evaluated using various metrics, such as accuracy, precision, recall, and F1 score. These metrics are used to determine how well the model can distinguish between individuals with SCI and healthy controls. The results of the testing process of different algorithms will be presented in the next chapter.

In some cases, the model's performance can be improved by adjusting its hyperparameters, such as the learning rate, regularization parameter, or number of

hidden layers in a neural network. Hyperparameter tuning involves searching for the optimal combination of hyperparameters that results in the best performance on the testing set. Moreover, to ensure that the model's performance is not influenced by the specific choice of training and testing data, cross-validation was used. Cross-validation involves dividing the data into multiple folds, training the model on one fold, and testing it on another. This process is repeated for each fold, and the performance metrics are averaged across all folds to obtain a more robust estimate of the model's performance.

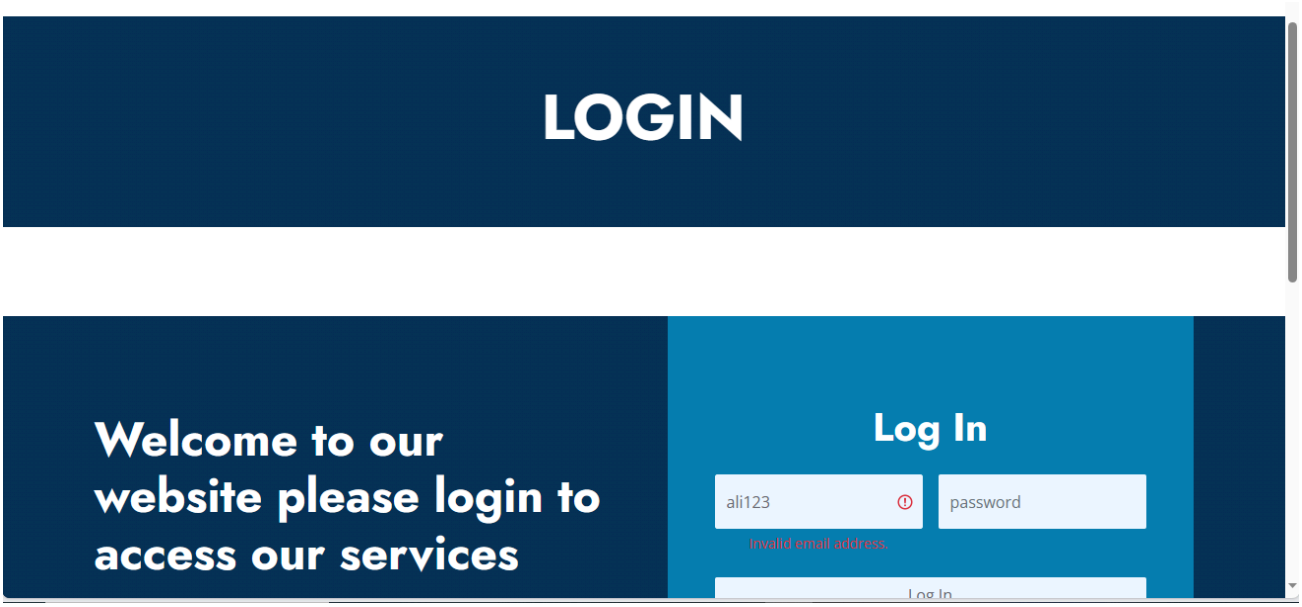
After testing and evaluating the performance of each algorithm, the best-performing model was selected for further analysis and validation. The selected model was used to classify new, unseen data and applied in clinical settings to improve diagnosis and treatment of SCI. In summary, the model training, testing, tuning and selection phases are critical steps in the classification of SCI using gene expression data and machine learning algorithms. They involve training and testing different models, evaluating their performance using various metrics, and selecting the best-performing model for further analysis and validation. By leveraging the power of these advanced techniques, accurate and reliable diagnostic tools for SCI have been developed and there is potential to improve outcomes for individuals with this debilitating condition.

5.3. Implementation of the Application & Testing

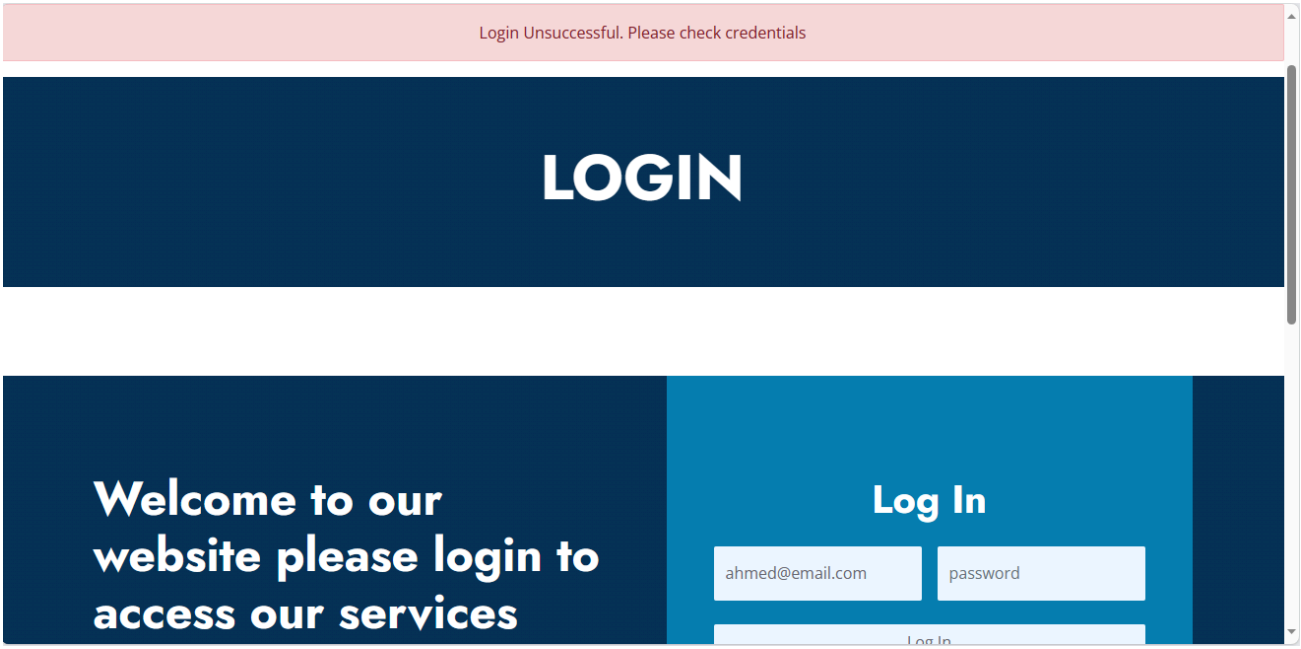
Snapshots of the system scenarios caused by the user

1- Invalid email and password

1.1 invalid email pattern



1.2 invalid email or password



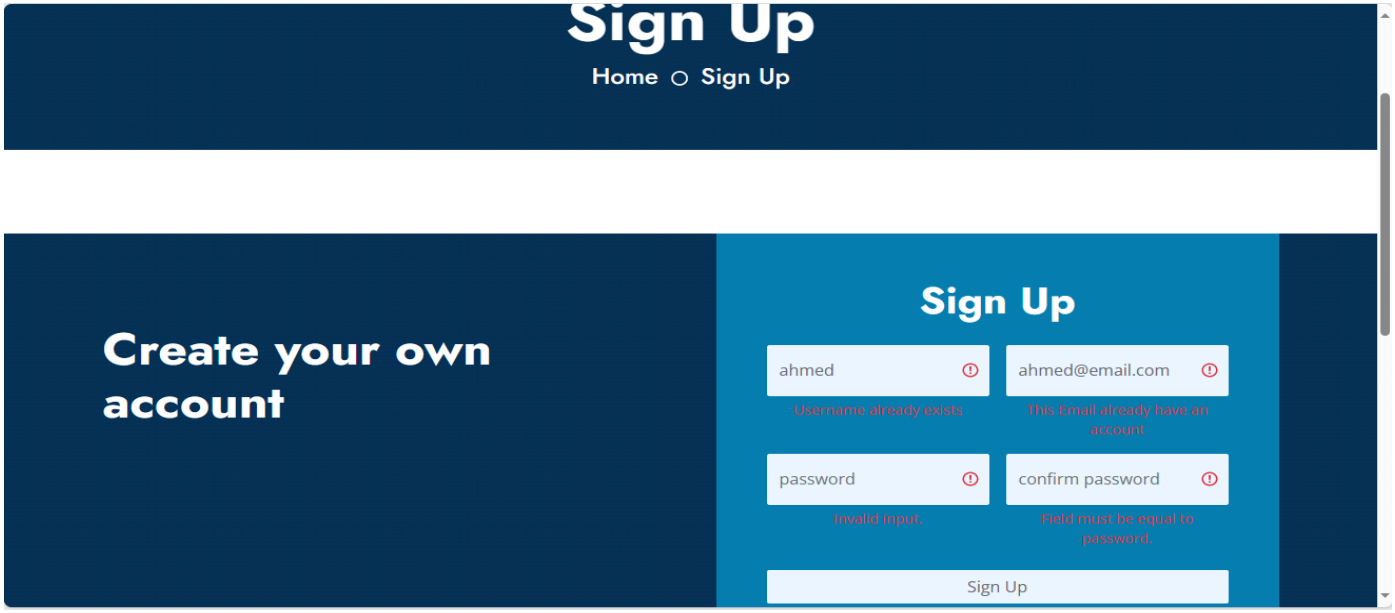
2- Invalid Register

2.1 .Username must be unique

2.2. Email must be unique

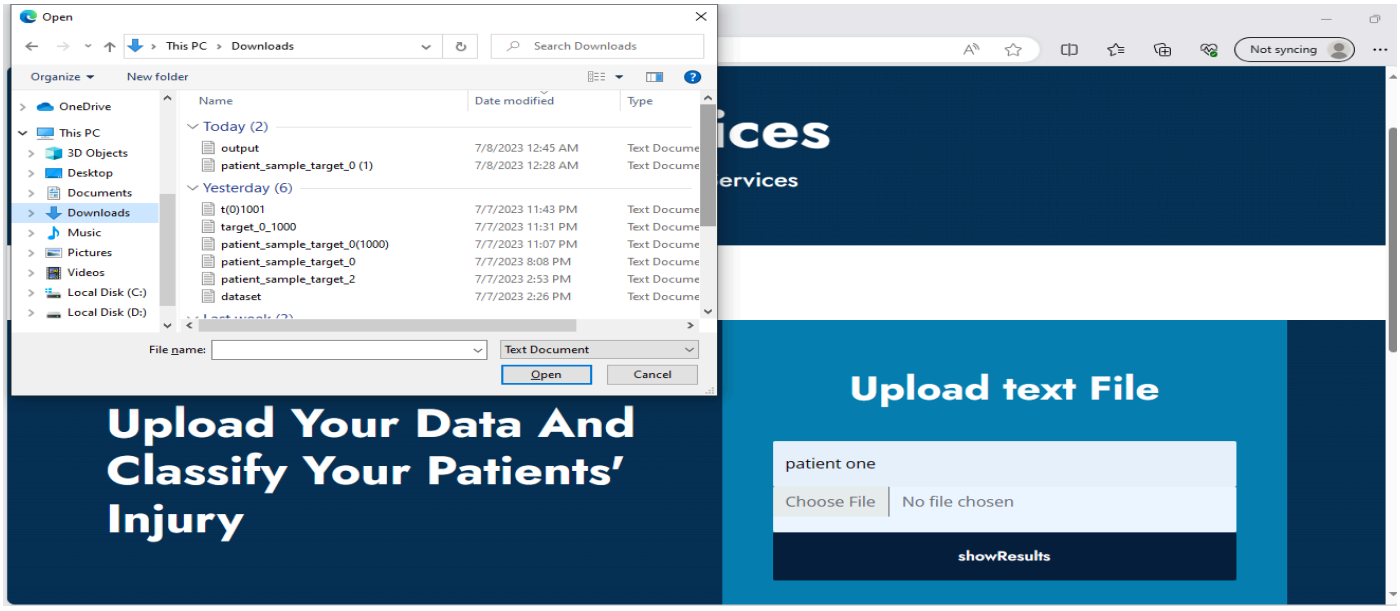
2.3. Password Restrictions is not Satisfying

2.4. Missmatching Password

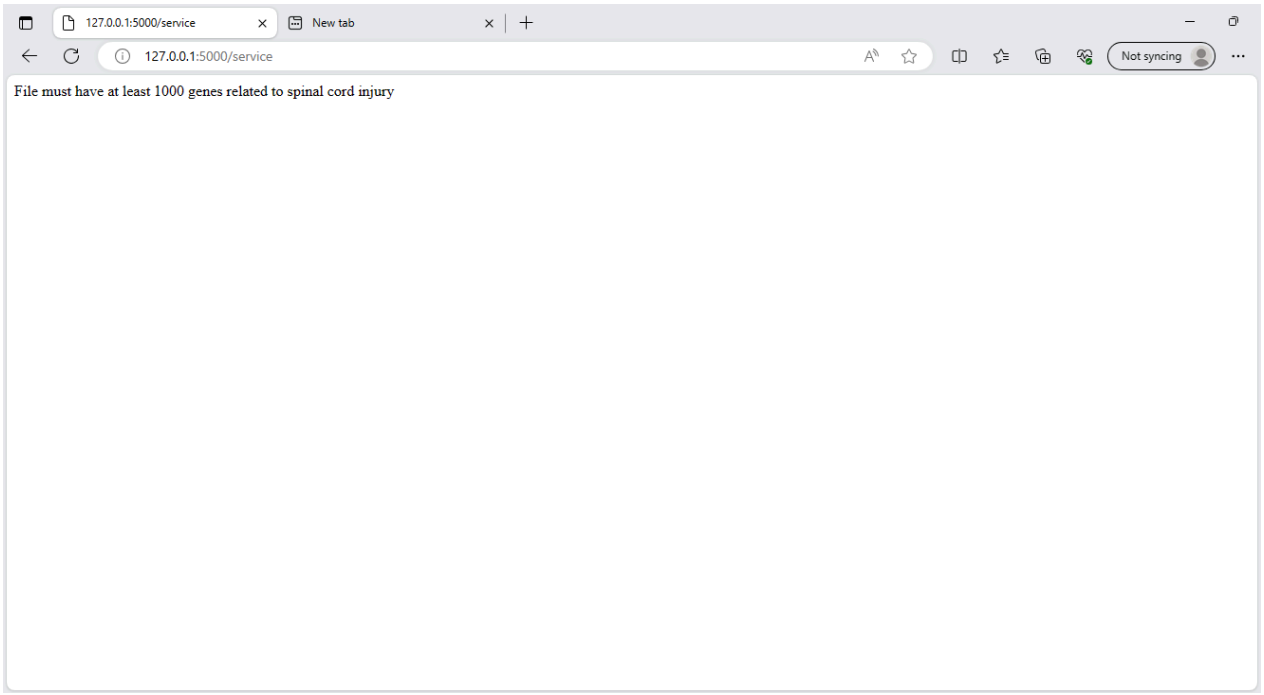


3- Upload Validation

3.1. Allowing The User To Upload Text File Only

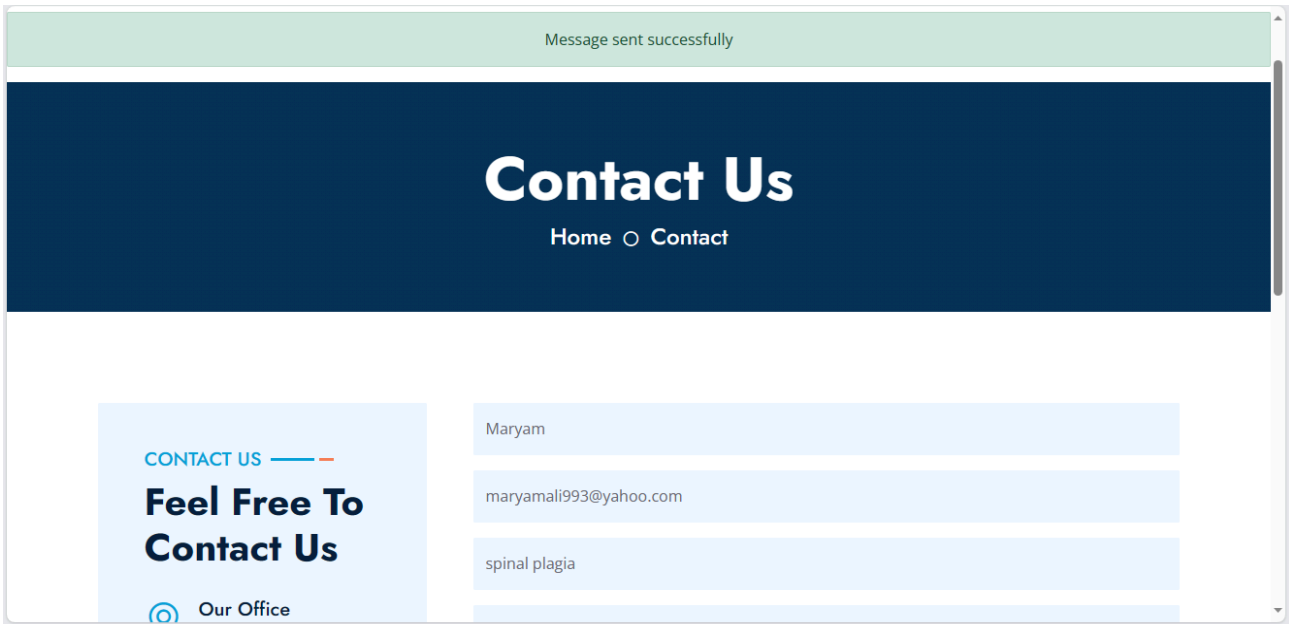


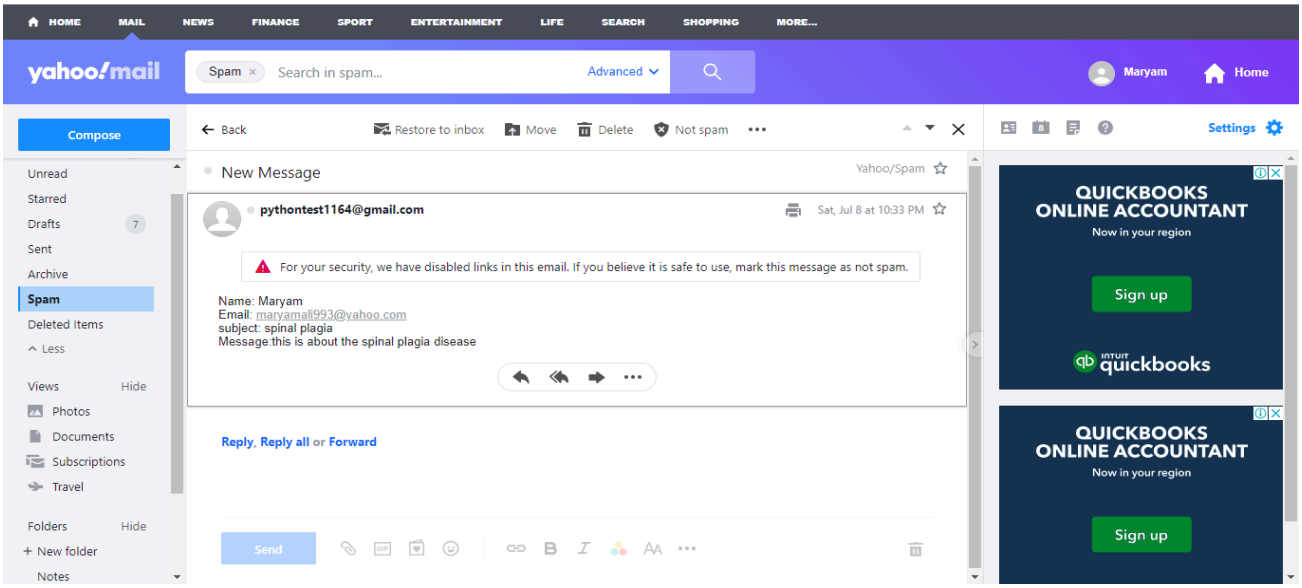
3.2 File Must Contain At Least 1000 Genes



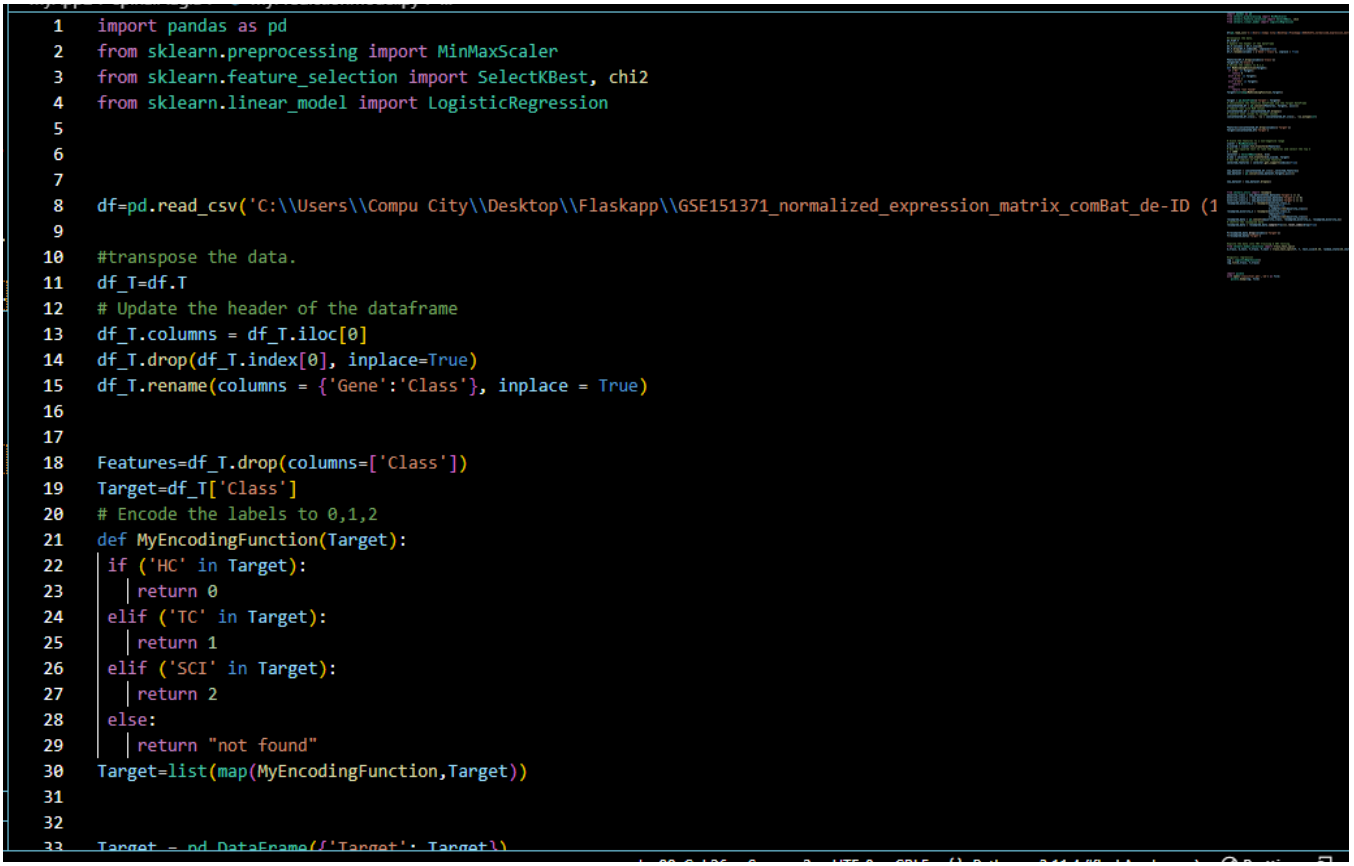
4- You Can Contact With Us

4.1. Website includes mail server

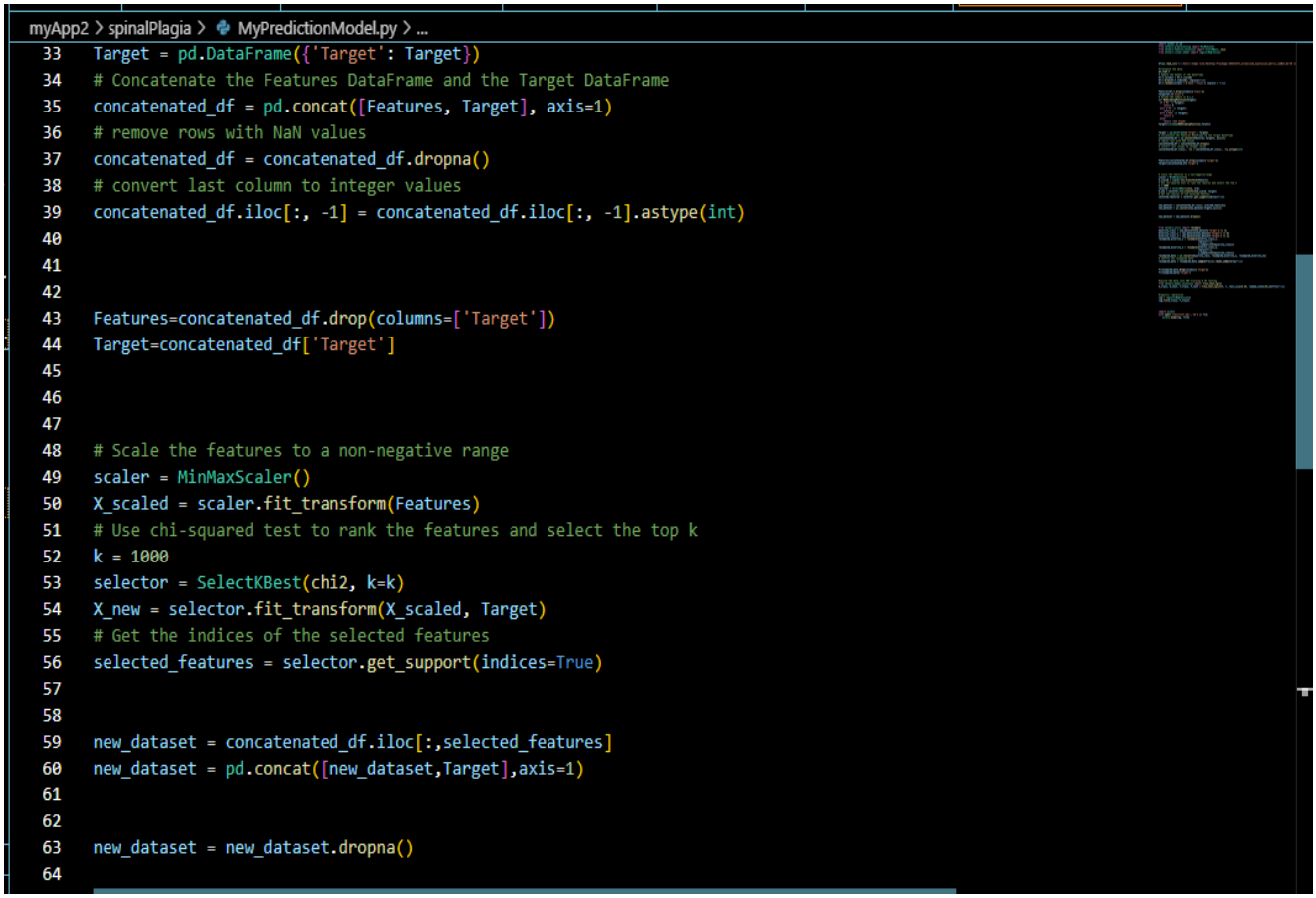




5-Our Model With PreProcessed Data



```python
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.linear_model import LogisticRegression



df=pd.read_csv('C:\\Users\\Compu City\\Desktop\\Flaskapp\\GSE151371_normalized_expression_matrix_comBat_de-ID (1

#transpose the data.
df_T=df.T
# Update the header of the dataframe
df_T.columns = df_T.iloc[0]
df_T.drop(df_T.index[0], inplace=True)
df_T.rename(columns = {'Gene':'Class'}, inplace = True)


Features=df_T.drop(columns=['Class'])
Target=df_T['Class']
# Encode the labels to 0,1,2
def MyEncodingFunction(Target):
 if ('HC' in Target):
   return 0
 elif ('TC' in Target):
   return 1
 elif ('SCI' in Target):
   return 2
 else:
   return "not found"
Target=list(map(MyEncodingFunction,Target))


Target = pd.DataFrame({'Target': Target})
```



```
myApp2 > spinalPlagia > MyPredictionModel.py > ...
33  Target = pd.DataFrame({'Target': Target})
34  # Concatenate the Features DataFrame and the Target DataFrame
35  concatenated_df = pd.concat([Features, Target], axis=1)
36  # remove rows with NaN values
37  concatenated_df = concatenated_df.dropna()
38  # convert last column to integer values
39  concatenated_df.iloc[:, -1] = concatenated_df.iloc[:, -1].astype(int)
40
41
42
43  Features=concatenated_df.drop(columns=['Target'])
44  Target=concatenated_df['Target']
45
46
47
48  # Scale the features to a non-negative range
49  scaler = MinMaxScaler()
50  X_scaled = scaler.fit_transform(Features)
51  # Use chi-squared test to rank the features and select the top k
52  k = 1000
53  selector = SelectKBest(chi2, k=k)
54  X_new = selector.fit_transform(X_scaled, Target)
55  # Get the indices of the selected features
56  selected_features = selector.get_support(indices=True)
57
58
59  new_dataset = concatenated_df.iloc[:,selected_features]
60  new_dataset = pd.concat([new_dataset,Target],axis=1)
61
62
63  new_dataset = new_dataset.dropna()
64
```

```python
from sklearn.utils import resample
majority_class = new_dataset[new_dataset['Target'] == 2]
minority_class_1 = new_dataset[new_dataset['Target'] == 0]
minority_class_2 = new_dataset[new_dataset['Target'] == 1]
resampled_minority_1 = resample(minority_class_1,
                                replace=True,
                                n_samples=len(majority_class))
resampled_minority_2 = resample(minority_class_2,
                                replace=True,
                                n_samples=len(majority_class))
resampled_data = pd.concat([majority_class, resampled_minority_1, resampled_minority_2])
# shuffle the resampled data
resampled_data = resampled_data.sample(frac=1).reset_index(drop=True)

F=resampled_data.drop(columns=['Target'])
T=resampled_data['Target']

#splite the data into 70% training & 30% testing.
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(F, T, test_size=0.30, random_state=20,shuffle=True)

#logistic regression
log = LogisticRegression()
log.fit(X_train, Y_train)



import pickle
with open('classifier.pkl','wb') as file:
    pickle.dump(log, file)
```

Chapter 6
Results

The results showed that the highest accuracy is achieved using logistic regression with feature selection and data augmentation, with an accuracy of 94.28% This indicates that the selected features and the augmented data helped to increase the accuracy of the model and improve its performance for the classification task.

In comparison, the accuracy of logistic regression with feature selection only was 83.33%, indicating that the use of data augmentation was beneficial for improving the model's performance. The accuracy of logistic regression without feature selection and augmentation was 83.33%, indicating that the feature selection and data augmentation techniques were necessary for achieving high accuracy on this classification task.These results highlight the importance of feature selection and data augmentation in improving the performance of classification models for gene expression data analysis. Feature selection helps to identify the most relevant features for the classification task, while data augmentation helps to increase the size and diversity of the training dataset, which can improve the generalization of the models.

The coming pages will contain the results of the test cases of the spinal cord injury detection models. These test cases were designed to evaluate the performance and accuracy of the classification models that were built using machine learning. The results will provide insights into how well the models are performing and how accurate they are at detecting spinal cord injury using gene expression data.

6.1. Accuracy without feature selection & Augmentation & scaling:

Models	Accuracy without feature selection & Augmentation & scaling
Logistic regression	Accuracy is 83.33%
Support vector machine	Accuracy is 83.33%
Decision tree	Accuracy is 66.66%
GaussianNB Naive Bayes	Accuracy is 66.66%
BernoulliNB Naive Bayes	Accuracy is 55.55%
KNN	(k=7):Accuracy is 55.55%
NN	Accuracy is 16.66%

6.2. Accuracy with feature selection only:

Models	Chi-Square	Select From Model	Variance Threshold
Logistic regression	Accuracy is 83.33%	Accuracy is 83.33%	Accuracy is 83.33%
Support vector machine	Accuracy is 83.33%	Accuracy is 83.33%	Accuracy is 83.33%
Decision tree	Accuracy is 72.22%	Accuracy is 77.77%	Accuracy is 77.77%
GaussianNB Naive Bayes	Accuracy is 72.22%	Accuracy is 72.22%	Accuracy is 72.22%
BernoulliNB Naive Bayes	Accuracy is 55.55%	Accuracy is 55.55%	Accuracy is 55.55%
KNN	(k=7): Accuracy is 72.22%	(k=7): Accuracy is 72.22%	(k=7): Accuracy is 72.22%
NN	Accuracy is 83.33%	Accuracy is 83.33%	Accuracy is 83.33%

6.3. Accuracy with Augmentation only:

Models	SMOTE	Over Sampled
Logistic regression	Accuracy is 83.33%	Accuracy is 83.33%
Support vector machine	Accuracy is 83.33%	Accuracy is 83.33%
Decision tree	Accuracy is 72.22%	Accuracy is 72.22%
GaussianNB Naive Bayes	Accuracy is 72.22%	Accuracy is 72.22%
BernoulliNB Naive Bayes	Accuracy is 55.55%	Accuracy is 55.55%
KNN	(k=7): Accuracy is 72.22%	(k=7): Accuracy is 72.22%
NN	Accuracy is 25.71%	Accuracy is 25.71%

6.4. Accuracy with SelectFromModel & (over_sampling || SMOTE):

Models	Accuracy with SelectFromModel & over_sampling	Accuracy with SelectFromModel & SMOTE
Logistic regression	Accuracy is 83.33%	Accuracy is 83.33%
Support vector machine	Accuracy is 83.33%	Accuracy is 83.33%
Decision tree	Accuracy is 66.66%	Accuracy is 77.77%
GaussianNB Naive Bayes	Accuracy is 72.22%	Accuracy is 72.22%
BernoulliNB Naive Bayes	Accuracy is 55.55%	Accuracy is 55.55%
KNN	(k=7): Accuracy is 72.22%	(k=7): Accuracy is 72.22%
NN	Accuracy is 40.00%	Accuracy is 40.00%

6.5. Accuracy with VarianceThreshold & (over_sampling || SMOTE):

Models	Accuracy with VarianceThreshold & over_sampling	Accuracy with VarianceThreshold & SMOTE
Logistic regression	Accuracy is 83.33%	Accuracy is 83.33%
Support vector machine	Accuracy is 83.33%	Accuracy is 83.33%
Decision tree	Accuracy is 72.22%	Accuracy is 72.22%
GaussianNB Naive Bayes	Accuracy is 72.22%	Accuracy is 72.22%
BernoulliNB Naive Bayes	Accuracy is 55.55%	Accuracy is 55.55%
KNN	(k=7): Accuracy is 72.22%	(k=7): Accuracy is 72.22%
NN	Accuracy is 40.00%	Accuracy is 40.00%

6.6. Accuracy for ChiSquare with scaling and (K=1000) & (resampling minority || SMOTE):

Models	Accuracy with ChiSquare & resampling minority	Accuracy with ChiSquare & SMOTE
Logistic regression	Accuracy is 85.71%	Accuracy is 85.71%
Support vector machine	Accuracy is 85.71%	Accuracy is 85.71%
Decision tree	Accuracy is 82.85%	Accuracy is 80.00%
GaussianNB Naive Bayes	Accuracy is 82.85%	Accuracy is 65.71%
BernoulliNB Naive Bayes	Accuracy is 85.71%	Accuracy is 77.14%
KNN	(k=7): Accuracy is 71.42%	(k=7): Accuracy is 77.14%
NN	Accuracy is 40.00%	Accuracy is 40.00%

6.7. Final table for Highest results of Models:

Models	Accuracy	Precession	Recall
Logistic Regression	0.85	0.87	0.85
SVM	0.85	0.87	0.85
BernoulliNB Naive Bayes	0.85	0.62	0.86

Chapter 7
Conclusions

In conclusion, spinal cord injury is a serious medical condition that can lead to permanent disability and a significant decrease in the quality of life. Early detection of SCI is important for improving patient outcomes. Machine learning algorithms represent a promising approach for detecting spinal cord injury from gene expression data. Our research project focused on building and evaluating several machine learning models for detecting non-traumatic SCI from human gene expression data. The results showed that logistic regression with feature selection and data augmentation achieved the highest accuracy of 85.71%, indicating that these techniques are important for improving the performance of classification models. Our web application provides a powerful tool for neurologists to diagnose and analyze SCI using gene expression data. Our study highlights the potential of machine learning algorithms for improving the detection and diagnosis of spinal cord injury and paves the way for further research in this area.

References

[1] Comparative analysis of molecular mechanism of spinal cord injury with time based on bioinformatics data | Spinal Cord (nature.com)

[2] A controlled spinal cord contusion for the rhesus macaque monkey - ScienceDirect

[3] GEO Accession viewer (nih.gov)

[4] RSNA CSF - Cervical Spine Fracture EDA | Kaggle