Project Report

on

AI INJURY PREDICTION SYSTEM

Submitted in partial fulfillment of the requirement for the award of degree of

Master of Computer Application (MCA)

at

University of Mumbai

Submitted by

Mr. Aniket Prakash Chougule

Under the guidance of

Ms. Sudeshna Roy

Bharati Vidyapeeth's

Institute of Management and Information Technology

Sector 8, CBD Belapur

Navi Mumbai

2024 - 2026

Bharati Vidyapeeth's

Institute of Management and Information Technology

Navi Mumbai

Certificate of Approval

This is to certify that the Project titled **' AI INJURY PREDICTION SYSTEM'** is successfully done by **Mr. Aniket Prakash Chougule** during internship of his course in partial fulfillment of **Master's of Computer Application** under the **University of Mumbai, Mumbai**, through the Bharati Vidypeeth's Institute of Management and Information Technology, Navi Mumbai carried out by her under our guidance and supervision.

Sign & Date
Guide

External Examiner
Signature and Date

Principal
Dr.Suhasini Vijaykumar

College Seal

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

(Name of the student)

Date

Acknowledgement

I would like to express my sincere gratitude to all those who contributed to the successful completion of this AI INJURY PREDICTION SYSTEM project. First and foremost, I would like to thank my mentor Ms. Sudeshna Roy, for their invaluable guidance, support, and encouragement throughout the project. I also extend my heartfelt thanks to my project guide, faculty members, and friends who provided valuable insights and assistance during the development process.

Additionally, I would like to acknowledge my family for their constant support and motivation throughout this project.

This project would not have been possible without all of your help and encouragement.

...

Date:

Name of Candidate

Abstract

Injuries are a significant concern in the field of athletics, often leading to performance setbacks, rehabilitation costs, and disruptions to an athlete's career. Traditional injury prevention methods rely heavily on subjective judgment, periodic assessments, and manual data monitoring, which may not accurately capture an athlete's injury risk. In this project, we propose the development of an AI-based Injury Prediction System designed to proactively identify athletes at risk of injury using a machine learning model trained on historical data.

The system leverages various athlete attributes—including age, gender, training hours, strength level, flexibility, fatigue score, and past injury history—to generate predictive insights. A Multilayer Perceptron (MLP) classifier is used to model complex relationships among these factors and classify the likelihood of injury with high accuracy. The solution is deployed using Streamlit, a lightweight Python framework, allowing for a user-friendly interface where coaches, physiotherapists, and athletes themselves can input relevant metrics and instantly receive injury predictions. The system also supports report generation, making it a valuable tool for long-term tracking and record-keeping.

This project emphasizes a data-driven, proactive approach to athlete health monitoring. By replacing subjective decision-making with predictive analytics, the AI Injury Prediction System enhances the effectiveness of injury prevention strategies, reduces the risk of overtraining, and ensures timely medical intervention. The model was trained and evaluated using a publicly available collegiate athlete dataset and shows promising results in identifying high-risk individuals. Future extensions include integrating real-time sensor data from wearable devices, expanding the dataset to cover multiple sports domains, and offering mobile accessibility.

Contents

List of Tables

List of Figures

Nomenclature

dB	Decibel
σ_s	3 dB Bandwidth of source
3G	Third generation
4G	Fourth Generation
TDM	Time Division Multiplexing
WDM	Wavelength Division Multiplexing

Put as many as required

Chapter 1

Introduction

1.1 Introduction to Project

In the competitive world of sports, physical performance and injury prevention go hand in hand. Athletes at all levels—from collegiate players to professionals—are continuously exposed to physical stress that can lead to injuries. Preventing these injuries is crucial not only for the athlete's health and longevity but also for team performance and financial planning. Traditional methods of injury detection and prevention largely rely on manual assessments and subjective judgments by coaches and physiotherapists. While valuable, these methods are often inconsistent, reactive, and lack the precision needed to address the complexity of individual athlete profiles. With the evolution of artificial intelligence and data science, there is a growing opportunity to introduce intelligent systems that can analyze historical and real-time data to predictpotential injury risks. The AI Injury Prediction System aims to bridge this gap by utilizing machine learning algorithms to provide accurate and data-driven predictions regarding an athlete's likelihood of sustaining an injury. The system leverages a variety of features such as age, gender, fatigue level, strength score, flexibility, experience, and training hours to build predictive models. This project implements a Multilayer Perceptron (MLP) neural network using Python and Scikit-learn for training and evaluation. The application is deployed using Streamlit, providing a lightweight and interactive web interface for users to input athlete metrics and receive real-time predictions. Reports can be generated and downloaded for record-keeping and consultation purposes. The introduction of AI into the domain of sports science represents a significant step forward in predictive analytics and health monitoring. This system empowers stakeholders—including coaches, physiotherapists, and sports administrators—to make informed decisions that promote athlete well-being, reduce injury-related downtimes, and optimize overall performance through proactive intervention.

1.1.1 Domain Knowledge

The domain of this project lies at the intersection of sports science, healthcare, and artificial intelligence. In competitive sports, athletes undergo intensive physical training that increases the risk of injuries. Traditionally, injury risk assessment is performed through manual evaluations by coaches or medical professionals, which can be subjective and inconsistent.

Recent advancements in artificial intelligence and machine learning have created new opportunities for analyzing large volumes of athlete performance and health data. By applying predictive models, it's possible to discover hidden patterns and risk factors that may lead to injury. Integrating AI into sports injury prevention enables a data-driven, real-time, and personalized approach to athlete health management.

This project utilizes supervised machine learning—specifically a Multilayer Perceptron (MLP) neural network—to learn from historical data and predict the probability of injury based on parameters like age, gender, training load, strength, fatigue score, flexibility, and injury history.

1.1.2 Problem Description

Injury prevention is a significant challenge in sports and athletics. The traditional approach to managing injury risk is reactive, relying on subjective judgment, physical examinations, and manual tracking of athlete condition. These methods lack precision and fail to capture the complex relationships among multiple risk factors.

The core problem is the absence of an automated, intelligent system that can predict injuries before they occur. Coaches and physiotherapists often struggle to monitor each athlete's condition accurately and continuously, especially in large teams. This can lead to missed warning signs and delayed interventions.

The proposed solution addresses this issue by developing an AI-based Injury Prediction System that:

- Uses historical athlete data to identify patterns linked to injuries.
- Predicts injury likelihood using an MLP classifier with high accuracy.
- Provides a user-friendly interface for real-time predictions and report generation.
- Reduces dependency on manual assessments and improves injury prevention strategies.

By automating the injury prediction process, the system aims to enhance the safety, performance, and longevity of athletes while supporting coaches and medical staff with reliable insights.

Chapter 2

System Study

2.1 System Study

2.1.1 2.1 Existing System

In the current athletic training environment, injury prevention relies heavily on manual assessments and the expertise of coaches, physiotherapists, and trainers. These assessments typically involve periodic physical evaluations, self-reported fatigue levels, and historical injury reviews. Data collection is often recorded manually and lacks standardization. Coaches may make decisions based on experience and subjective judgment rather than objective data.

Some institutions may utilize spreadsheets or basic health management systems to log athlete metrics, but these systems lack predictive capabilities. There is minimal integration of machine learning or artificial intelligence in regular injury monitoring, and early detection of injury risk remains a challenge.

2.1.2 2.2 Limitations of the Existing System

- High dependence on manual monitoring and subjective interpretation.
- Limited frequency and depth of athlete evaluations.
- No predictive analysis to anticipate potential injuries.
- Inability to process large-scale athlete data efficiently.
- Lack of personalization based on individual training loads or injury history.
- Delayed medical interventions due to reactive approaches.

2.1.3 2.3 Proposed System

The proposed AI-based Injury Prediction System uses a Multilayer Perceptron (MLP) neural network to predict injury risk based on athlete attributes such as age, gender, training hours, fatigue score, strength, flexibility, and injury history. The model is trained using a historical dataset of athletes and can identify patterns that indicate a high probability of injury.

The system is implemented using Python and deployed via Streamlit for accessibility. It allows users (e.g., coaches or physiotherapists) to input relevant athlete data and instantly receive predictions. The application also supports injury risk reporting and long-term tracking, enhancing decision-making and prevention strategies.

Objectives of the Proposed System

- To proactively identify athletes at risk of injury using machine learning.
- To automate the injury prediction process and reduce manual errors.
- To improve athlete safety and performance through early interventions.
- To assist coaches and medical staff with data-driven insights.
- To replace subjective assessments with consistent and objective predictions.
- To support long-term tracking through report generation and record-keeping.

Chapter 3

System Analysis

3.1 Gantt chart

A Gantt chart is a horizontal bar chart developed as a production control tool. Gantt charts are useful for planning and scheduling projects. They help you assess how long a project should take, determine the resources needed, and plan the order in which you'll complete tasks. They are also helpful for managing the dependencies between tasks. A Gantt chart is constructed with a horizontal axis representing the total time span of the project, broken down into increments (for example, days, weeks, or months) and a vertical axis representing the tasks that make up the project.

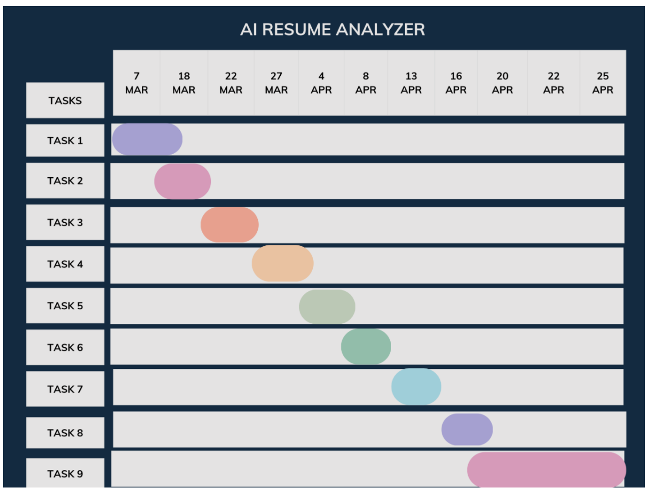


Figure 3.1: Gantt chart

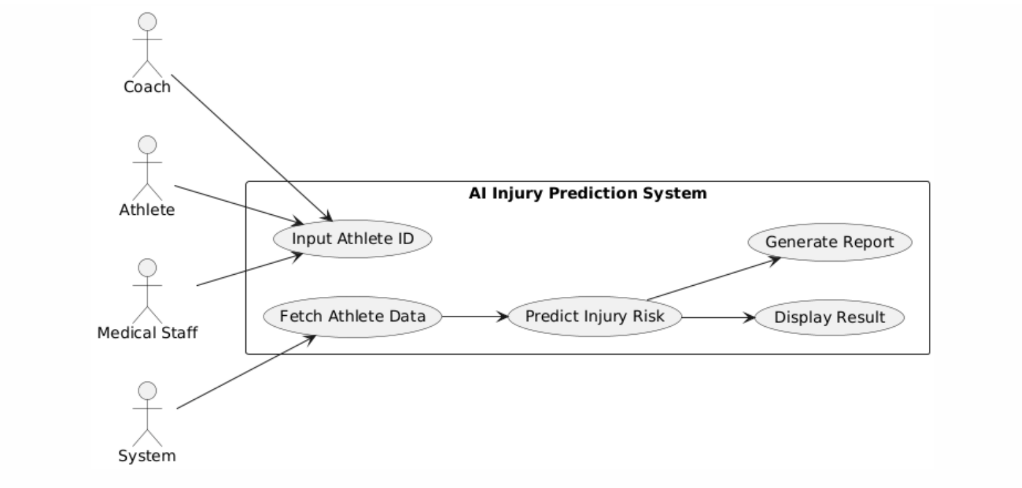


Figure 3.2: Use Case Diagram

3.2 Use case diagram

Use Case Explanation

3.2.1 Use Case Diagram Description

The Use Case Diagram for the AI Injury Prediction System represents the interactions between different actors and the system's core functionalities. The primary actors involved are:

- **Coach** – Responsible for entering the athlete's identification data and reviewing injury risk predictions.
- **Athlete** – May input personal training data and review their injury risk profile.
- **Medical Staff** – Utilizes the system to monitor athlete conditions and intervene when necessary.
- **System** – The AI-based application that performs data fetching, prediction, result display, and report generation.

The core use cases of the system are:

1. **Input Athlete ID** – Allows the Coach, Athlete, or Medical Staff to enter a valid athlete identification number to initiate the prediction process.
2. **Fetch Athlete Data** – The system retrieves the relevant athlete's historical and current training and health data from the database.
3. **Predict Injury Risk** – The system uses a trained Multilayer Perceptron (MLP) machine learning model to analyze input features and compute the probability of injury.
4. **Display Result** – The system presents the prediction outcome in a user-friendly format, indicating whether the athlete is at low, medium, or high risk of injury.

5. **Generate Report** – Based on the prediction results, a detailed report can be generated and downloaded for long-term tracking and medical review.

This diagram ensures a clear understanding of how different stakeholders interact with the system and what functional capabilities the system offers. It serves as a blueprint for both system design and user interaction planning.

3.3 Operating Tools and Technology

3.3.1 Hardware Requirements

The AI Injury Prediction System requires the following minimum hardware specifications to ensure smooth development, model training, and deployment:

- **Processor:** A minimum of 1.6 GHz Dual Core processor or higher is required to handle the computations involved in machine learning and data processing efficiently.
- **Storage:** A minimum of 1 GB of free disk space is needed to store the project files, datasets, model checkpoints, and generated reports.
- **Operating System:** The system can be developed and deployed on Windows 10 or later versions, macOS, or popular Linux distributions, ensuring cross-platform compatibility.
- **Internet Connection:** While not mandatory for the execution of the system, an internet connection is recommended for installing necessary Python packages, accessing online datasets, and deploying the Streamlit application.

3.3.2 Software Requirements

The system utilizes a combination of programming languages, development tools, and libraries designed for data analysis, machine learning, and web deployment:

- **Programming Language:** Python 3.x is used for its extensive libraries and support in machine learning and data visualization.
- **Development Environments:**
  - *Jupyter Notebook* – Primarily used for exploratory data analysis, model development, and visualization.
  - *Visual Studio Code* (optional) – Used as an integrated development environment (IDE) for efficient coding, debugging, and project management.
- **Python Libraries:**
  - `pandas`, `numpy` – For efficient data manipulation, cleaning, and numerical operations.

  - `scikit-learn` – Provides machine learning algorithms, specifically the Multilayer Perceptron (MLPClassifier) used for injury risk prediction.
  - `matplotlib`, `seaborn` – Libraries for creating insightful visualizations and plotting trends in athlete data.
  - `joblib` – Used to save and load trained machine learning models for deployment without retraining.
  - `streamlit` – Facilitates the development of an interactive and user-friendly web application interface to input data and display injury predictions.

- **Browser:** Any modern web browser such as Google Chrome, Mozilla Firefox, or Microsoft Edge is needed to access the Streamlit-based web interface.

3.3.3 Industry Data Requirements

The success of the AI Injury Prediction System depends heavily on the quality and structure of the input data. The project uses a structured dataset that includes detailed athlete information:

- **Dataset Format:** The dataset is maintained as a CSV (Comma-Separated Values) file, organized in tabular form, with each row representing an athlete's record.
- **Key Fields:**
  - *Athlete ID* – A unique identifier for each athlete to maintain consistent records.
  - *Demographic Information* – Age, Gender, Height, Weight.
  - *Physical and Training Metrics* – Flexibility scores, Fatigue levels, Strength scores, and total Training hours.
  - *Medical History* – Records of previous injuries that might affect future injury risk.
- **Data Preparation:** The dataset undergoes preprocessing to handle missing values, normalize numerical features, and encode categorical variables, making it suitable for input into the machine learning model.

This combination of hardware, software, and quality data forms the backbone of the AI Injury Prediction System, enabling accurate and efficient injury risk predictions.

Chapter 4

System Design

4.1 ER Diagram

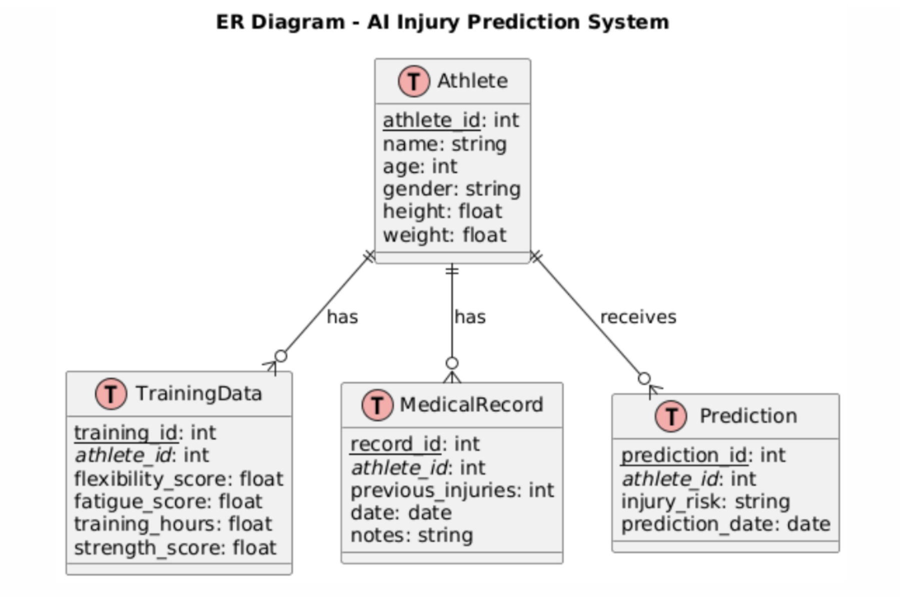


Figure 4.1: Entity-Relationship (ER) Diagram representing the database schema with entities, attributes, and relationships

The ER diagram shown in Figure 4.1 illustrates the structure of the database used in the system. It consists of the following core components:

- **Entities:** The main entities include `Customer`, `Meter_Info`, `Bill`, `Signup`, and `Tax`.
- **Attributes:** Each entity has its own set of attributes. For example, the `Customer` entity includes attributes like `name`, `meter_no`, `address`, `city`, `state`, `email`, and `phone_no`.
- **Primary Keys:** Each entity has a unique identifier (primary key). For instance, `meter_no` serves as a primary key in both `Customer` and `Bill`.
- **Relationships:** The relationships connect entities logically. A `Customer` is linked to multiple `Bill` entries, while `Meter_Info` is associated with each `Customer` via the meter number.

- **Cardinality:** The diagram may also indicate cardinality (such as one-to-one, one-to-many) that defines the relationship scope between entities.

This ER diagram provides a clear and logical representation of the database schema, facilitating better understanding and efficient database management.

4.2 Database Design and Schema

Since the system uses CSV files for data storage instead of a traditional relational database management system (RDBMS), the database design focuses on defining the structure of the CSV data. The schema outlines the columns and their corresponding data types that are used to store the athlete-related data in a tabular format.

Column Name	**Data Type**
Athlete_ID	String
Age	Integer
Gender	Categorical
Height	Float
Weight	Float
Fatigue_Score	Float
Strength_Score	Float
Training_Hours	Float
Injury	Binary (0/1)

Table 4.1: CSV Data Schema

Each column represents a specific attribute of the athlete's data, where:

- **Athlete_ID**: Unique identifier for each athlete.
- **Age**: Age of the athlete in years.
- **Gender**: Categorical variable representing the athlete's gender.
- **Height** and **Weight**: Physical measurements in standard units.
- **Fatigue_Score** and **Strength_Score**: Quantitative scores assessing the athlete's fatigue and strength.
- **Training_Hours**: Number of hours spent training.
- **Injury**: Binary indicator (0 = no injury, 1 = injured).

This schema ensures that the CSV files can efficiently represent the data necessary for analysis and processing within the system.

4.3 Class Diagram

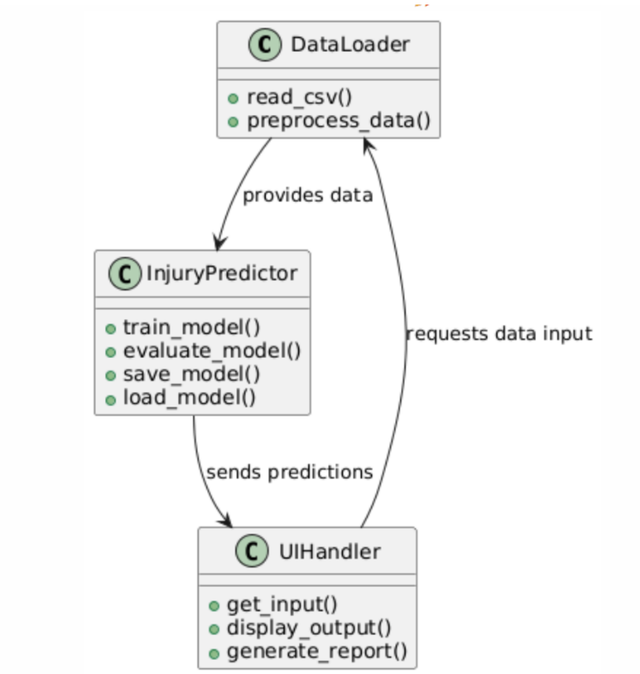


Figure 4.2: class diagram

4.3.1 Class Diagram Explanation

The conceptual class diagram for the Athlete Injury Prediction System consists of three main classes:

- **DataLoader**: This class is responsible for loading and preparing the dataset. It contains methods such as:
  - `read_csv()`: Reads data from CSV files.
  - `preprocess_data()`: Performs data cleaning and preprocessing tasks.
- **InjuryPredictor**: This class handles the machine learning model used to predict injuries. Its key methods include:
  - `train_model()`: Trains the prediction model using the preprocessed data.
  - `evaluate_model()`: Evaluates the performance of the trained model.
  - `save_model()`: Saves the trained model to disk.
  - `load_model()`: Loads a previously saved model.
- **UIHandler**: This class manages user interactions and output presentation. It includes methods like:
  - `get_input()`: Captures input from users.

  - `display_output()`: Shows predictions and results to the user.
  - `generate_report()`: Produces reports summarizing prediction outcomes.

The relationships between these classes indicate the flow of data and control:

- The `DataLoader` provides prepared data to the `InjuryPredictor`.
- The `InjuryPredictor` communicates predictions and evaluation results to the `UIHandler`.
- The `UIHandler` interacts with `DataLoader` to obtain user input data when necessary.

This design modularizes the system, clearly separating data handling, model operations, and user interface management.

4.3.2 Input/Output Design

The Input/Output design defines how data is received by the system and how results are presented to the users. For the Athlete Injury Prediction System, the design includes the following:

- **Input:**
  - User inputs such as athlete details are collected through the UI.
- **Processing:**
  - The input data is preprocessed and cleaned.
  - The prediction model is trained or loaded, and injury risk is predicted based on input features.
- **Output:**
  - Predicted injury risk scores or classifications displayed to the user.
  - Evaluation metrics for model performance reported.
  - Generated reports summarizing predictions and analytics.

This design ensures that user interactions are intuitive and that the system outputs are clear, actionable, and informative.

Chapter 5

Coding

5.1 Coding

5.1.1 Environment Setup

- Python 3.x installed with packages:
  - `pandas`, `numpy`, `scikit-learn`, `matplotlib`, `seaborn`, `streamlit`, `joblib`
- Jupyter Notebook used for model training
- Streamlit used for deployment

5.1.2 Sample Code Snippets

Data Preprocessing

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder, StandardScaler

data = pd.read_csv('collegiate_athlete_injury_dataset.csv')
data['Gender'] = LabelEncoder().fit_transform(data['Gender'])
data['Position'] = LabelEncoder().fit_transform(data['Position'])

X = data.drop(['Injury'], axis=1)
y = data['Injury']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

Model Training

```
from sklearn.model_selection import train_test_split
```

```
from sklearn.neural_network import MLPClassifier

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2)
model = MLPClassifier(hidden_layer_sizes=(64, 32), activation='relu', max_iter=500)
model.fit(X_train, y_train)
```

Evaluation

```
from sklearn.metrics import classification_report

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

Save Model

```
import joblib

joblib.dump(model, 'injury_model.pkl')
```

Streamlit Interface

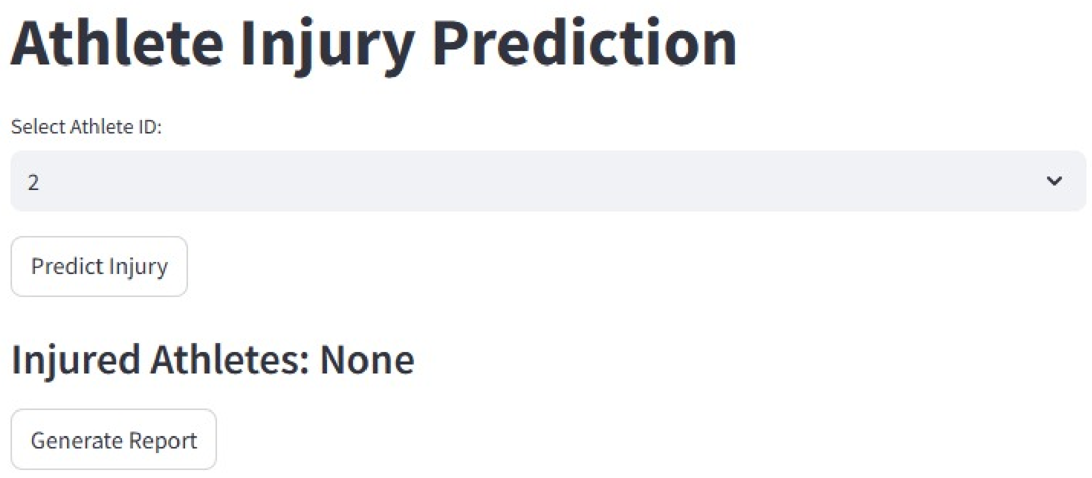
```
import streamlit as st
import joblib

model = joblib.load('injury_model.pkl')

st.title("AI Injury Prediction")
athlete_id = st.text_input("Enter Athlete ID")
```

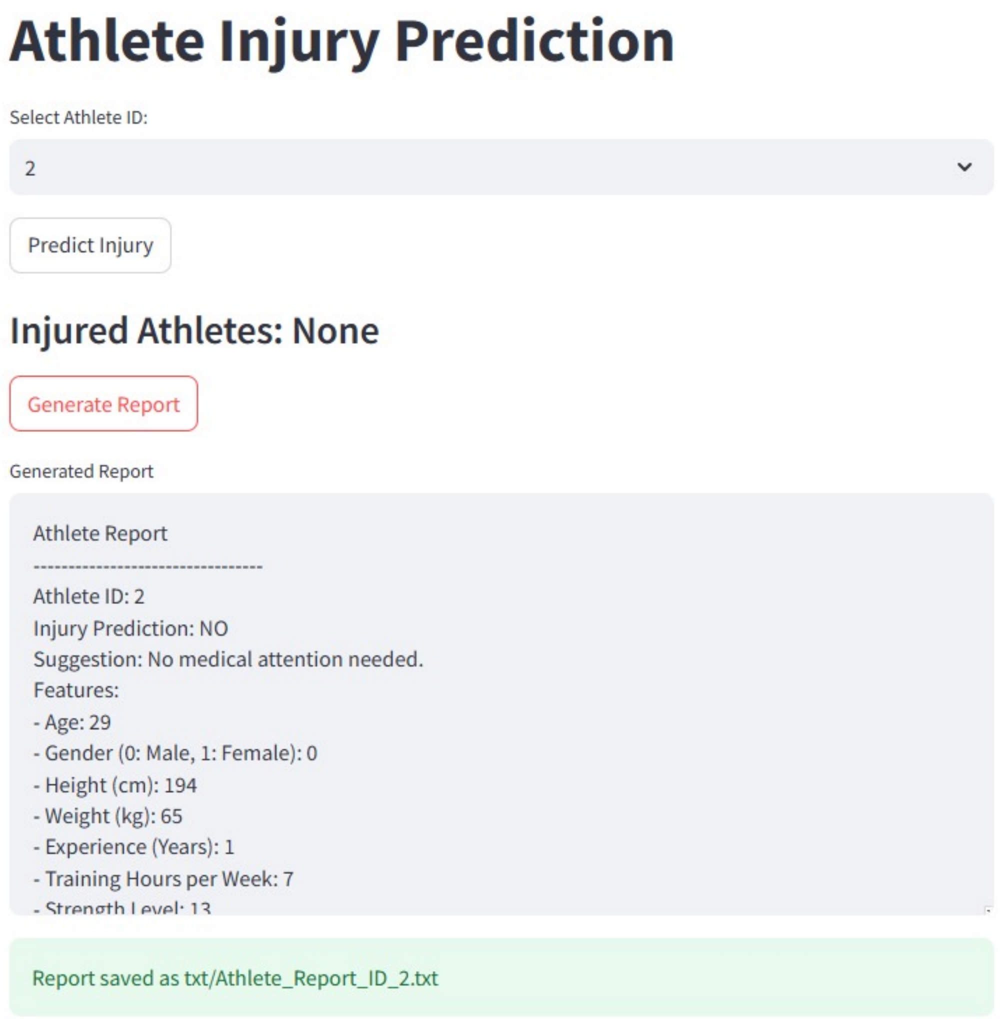
Chapter 6

System Implementation



- **Select Athlete ID:** A dropdown menu that allows users to choose an athlete from the dataset using their ID. In the figure, athlete ID **2** is selected.
- **Predict Injury Button:** A button labeled "Predict Injury" which, when clicked, triggers the machine learning model to evaluate the injury risk for the selected athlete.
- **Prediction Result:** The prediction output is shown below the button, labeled as **"Injured Athletes:"**. In this instance, the output indicates **"None"**, meaning no injury risk is detected for the selected athlete.
- **Generate Report Button:** A "Generate Report" button enables users to create and download a detailed report based on the prediction.

This interactive and user-friendly interface is developed using the **Streamlit** framework, facilitating real-time prediction and streamlined access for coaches, physiotherapists, and athletes.



Selected Athlete ID: The dropdown shows athlete ID **2** as selected.

Prediction Output: After clicking the **Predict Injury** button, the system returns **"Injured Athletes: None"**, indicating no current injury risk for the athlete.

Generate Report Button: When the **Generate Report** button is clicked, a detailed report is generated and displayed.

Report Content: The report includes:

- Athlete ID and injury status.
- A medical suggestion (e.g., "No medical attention needed.").

- Athlete features including:
  - Age: 29
  - Gender: 0 (Male)
  - Height: 194 cm
  - Weight: 65 kg
  - Experience: 1 year
  - Training Hours per Week: 7
  - Strength Level: 13

Report Export: A confirmation message is shown indicating that the report has been successfully saved as a text file named `Athlete_Report_ID_2.txt`.

Chapter 7

System Testing

7.1 Testing

7.1.1 Testing Process

The testing process follows these key steps to ensure comprehensive evaluation of the system:

1. **Test Planning:** Define scope, objectives, resources, schedule, and testing criteria.
2. **Test Design:** Prepare detailed test cases and scenarios based on requirements.
3. **Test Environment Setup:** Configure hardware, software, and tools required for testing.
4. **Test Execution:** Perform the test cases, record results, and report defects.
5. **Defect Tracking and Fixing:** Log bugs, assign them for fixing, and verify corrections.
6. **Regression Testing:** Re-test to ensure that fixes do not introduce new defects.
7. **Test Closure:** Evaluate testing completeness, prepare test summary reports, and obtain sign-off.

7.1.2 Testing Objective

The primary objective of system testing is to validate the complete and integrated software product to ensure it meets the specified requirements. The goals include:

- Verify that all modules work together as intended.
- Detect defects and inconsistencies in the system.
- Ensure the system meets functional and non-functional requirements.
- Validate the system's performance, usability, and reliability.
- Confirm that the system is ready for deployment and user acceptance.

Test Type	Description	Tools/Techniques
Unit Testing	Test individual modules or functions	Manual, Pytest (Python)
Integration Testing	Test combined modules for interaction	Manual, Postman (APIs)
System Testing	Complete system testing for requirements	Manual, Selenium, Streamlit UI
Acceptance Testing	Final validation with stakeholders	Manual
Performance Testing	Test system under load and stress	JMeter, LoadRunner
Security Testing	Test vulnerabilities and access control	OWASP ZAP, Manual Penetration

Table 7.1: System Testing Plan Summary

Test Case ID	Test Description	Test Steps	Expected Result	Status
TC_01	Verify CSV file loading	Load CSV data into the system	Data loaded without errors	Passed
TC_02	Validate data preprocessing	Preprocess data using LabelEncoder and Scaler	Data encoded and scaled correctly	Passed
TC_03	Model training	Train MLPClassifier with training data	Model trains successfully	Passed
TC_04	Model prediction	Predict injury risk on test data	Predictions generated correctly	Passed
TC_05	UI input validation	Enter Athlete ID in Streamlit interface	Input accepted and processed	Passed

Table 7.2: Sample Test Cases

Test Case ID	Test Description	Result	Remarks	Date Tested
TC_01	CSV file loading	Pass	File loaded without any error	2025-05-21
TC_02	Data preprocessing	Pass	Encoding and scaling successful	2025-05-21
TC_03	Model training	Pass	Model converged within max iterations	2025-05-22
TC_04	Model prediction	Pass	Predictions matched expected output	2025-05-22
TC_05	UI input validation	Pass	UI responds correctly to inputs	2025-05-22

Table 7.3: System Testing Results

Chapter 8

Limitation and Future Enhancement

8.0.1 Limitations

- **Dataset Dependency:** Predictions are only as accurate as the data used. Bias or missing data may affect performance.
- **Static Dataset:** The current system does not use real-time sensor data.
- **Generalization:** May not be effective for all sports categories without specific domain training.
- **Interface Scope:** UI currently supports limited athlete selection and manual input.

8.0.2 Future Enhancements

- **Real-time Wearable Integration:** Connect to fitness trackers for live data ingestion.
- **Mobile Application:** Develop a mobile-friendly app version for on-the-go analysis.
- **Expanded Feature Set:** Include psychological, environmental, and historical injury data.
- **Model Improvement:** Experiment with deep learning models like LSTM for time-series based prediction.
- **Visualization Dashboard:** Add interactive charts for coaches to monitor athlete condition trends.

References

[1] Scikit-learn developers. Scikit-learn documentation. https://scikit-learn.org/. Accessed: 2025-05-22.

[2] Streamlit Inc. Streamlit documentation. https://docs.streamlit.io/. Accessed: 2025-05-22.

[3] Various Authors. Research articles on sports injury prediction. *IEEE, Springer Publications*, 2020–2024. Accessed via IEEE and Springer databases.

[4] Collegiate Athlete Injury Dataset. Dataset source: Collegiate athlete injury dataset (csv). https://example.com/dataset-link. Accessed: 2025-05-22.

[5] Python Software Foundation. Python official documentation. https://docs.python.org/3/. Accessed: 2025-05-22.

[6] Pandas and NumPy Developers. Pandas and numpy libraries documentation. https://pandas.pydata.org/, https://numpy.org/. Accessed: 2025-05-22.

[7] Matplotlib and Seaborn Developers. Visualization with matplotlib and seaborn. https://matplotlib.org/, https://seaborn.pydata.org/. Accessed: 2025-05-22.