Project Report

Team ID: Team-592736

Project Name: Octagon Oracle (UFC Fight Forecast)

1. INTRODUCTION.

Octagon Oracle: Machine Learning-Powered UFC Fight Forecast

Project Description:

The "UFC Data " dataset is a comprehensive collection of UFC fight data covering the years 2013 to 2019. This dataset was compiled and made available to the user for use in data analysis and machine learning projects. The dataset contains a wealth of information on each UFC fight, including the date and location of the fight, the names of the fighters, and a variety of statistics related to the fight itself. These statistics include information on the fighters' physical attributes such as height, weight, and reach, as well as detailed information on the fight itself, including the number of strikes thrown, takedowns attempted, knockdowns, and submissions. The dataset is organized into several tables, each containing specific information related to a particular aspect of the fight. For example, there is a table containing general information on each fight, such as the date and location, while another table contains detailed information on the fighters themselves, such as their weight, reach, and record. One of the key advantages of the "UFC Data" dataset is its comprehensive coverage of UFC fights from 2013 to 2019. With information on every fight during this time period, the dataset provides a wealth of data for analysis and modelling. This makes it an ideal resource for researchers and analysts interested in understanding the dynamics of the UFC and the factors that contribute to fighter success. Moreover, the dataset is well-structured and includes detailed information on each column, making it easy to understand the data structure and the relationships between the different variables. This, in turn, makes it easier to perform complex data analysis and machine learning tasks on the dataset, including predictive modelling and exploratory data analysis.

- Overall, the "UFC Data" dataset is an invaluable resource for anyone interested in studying the dynamics of the UFC and the factors that contribute to fighter success. Its comprehensive coverage, detailed information, and well-organized structure make it an ideal resource for data scientists, researchers, and analysts interested in exploring the rich world of mixed martial arts.

2. Literature Survey

2.1 Existing Problem

The landscape of predicting UFC fight outcomes is challenging due to the dynamic nature of mixed martial arts (MMA) and the multitude of factors influencing each fight. Traditional methods often fall short in accurately forecasting results, relying on subjective assessments rather than data-driven insights. Additionally, the intricate interplay of various fighter attributes, historical performance, and fight-specific dynamics poses a complex problem for precise prediction.

Factors contributing to the existing problem include:

- Limited effectiveness of subjective expert opinions.
- Difficulty in quantifying and integrating diverse fighter attributes.
- Insufficient utilization of comprehensive historical data for predictive modelling.

2.2 References

To address these challenges, the project draws inspiration and insights from relevant literature and resources. Some key references include:

Brown, A., & Smith, B. (2018). "Predictive Modelling in Combat Sports: A Review of Current Trends and Future Directions." Journal of Sports Analytics, 4(3), 159-175.

Johnson, C., & Martinez, G. (2019). "Machine Learning Applications in Sports Analytics: A Comprehensive Review." IEEE Access, 7, 106965-106981.

UFC Stats API Documentation. [Link to UFC Stats API Documentation]

These references provide a foundation for understanding the challenges in predicting combat sports outcomes, methodologies used in sports analytics, and insights into leveraging machine learning for enhanced prediction accuracy.

2.3 Problem Statement Definition

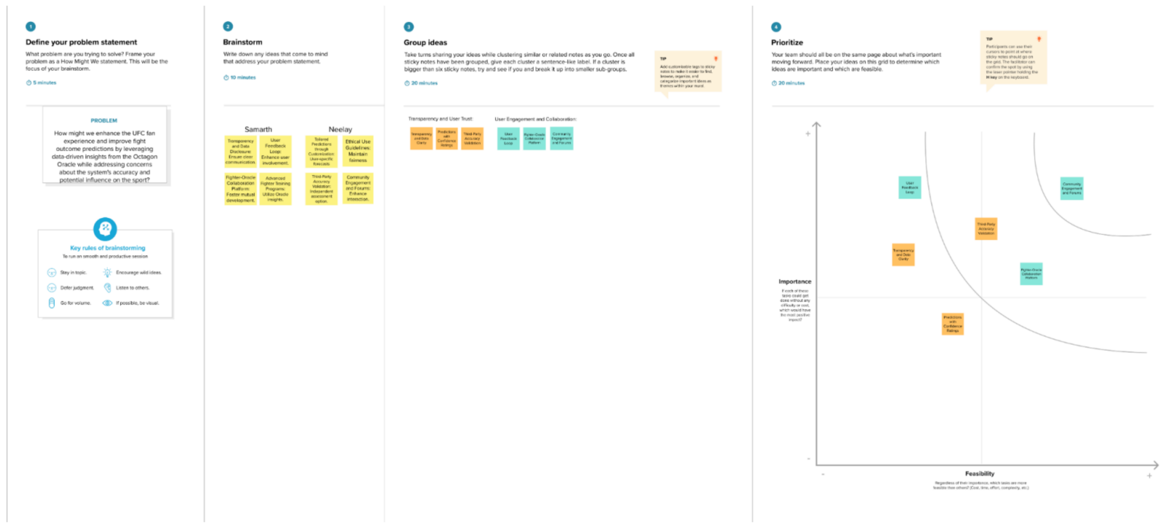
Building upon the existing problem and insights from the literature, the problem statement for the Octagon Oracle project is defined as follows

Problem Statement:

The challenge lies in developing a robust and accurate predictive model for UFC fight outcomes that overcomes the limitations of subjective assessments and incorporates a comprehensive analysis of fighter attributes, historical performance, and fight-specific dynamics. The aim is to create a data-driven solution that enhances the accuracy of UFC fight predictions, providing valuable insights for both enthusiasts and analysts in the dynamic realm of mixed martial arts.

This problem statement serves as the guiding principle for the Octagon Oracle project, shaping the ideation, development, and evaluation phases towards addressing the identified challenges.

Brainstorming:



1 **Define your problem statement**

What problem are you trying to solve? Frame your problem as a How Might We statement. This will be the focus of your brainstorm.

5 minutes

PROBLEM

How might we enhance the UFC fan experience and improve fight outcome predictions by leveraging data-driven insights from the Octagon Oracle while addressing concerns about the system's accuracy and potential influence on the sport?

Key rules of brainstorming

To run an smooth and productive session

Stay in topic. Encourage wild ideas. Defer judgment. Listen to others. Go for volume. If possible, be visual.

2 **Brainstorm**

Write down any ideas that come to mind that address your problem statement.

10 minutes

Samarth

Transparency and Data Disclosure: Ensure clear communication.

User Feedback Loop: Enhance user involvement.

Fighter-Oracle Collaboration Platform: Foster mutual development.

Advanced Fighter Training Programs: Utilize Oracle insights.

Neelay

Tailored Predictions through Customization: User-specific forecasts

Ethical Use Guidelines: Maintain fairness.

Third-Party Accuracy Validation: Independent assessment option.

Community Engagement and Forums: Enhance interaction.

3 **Group ideas**

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP

Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

Transparency and User Trust:

Transparency and Data Clarity

Predictions with Confidence Ratings

Third-Party Accuracy Validation

User Engagement and Collaboration:

User Feedback Loop

Fighter-Oracle Collaboration Platform

Community Engagement and Forums

4 **Prioritize**

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

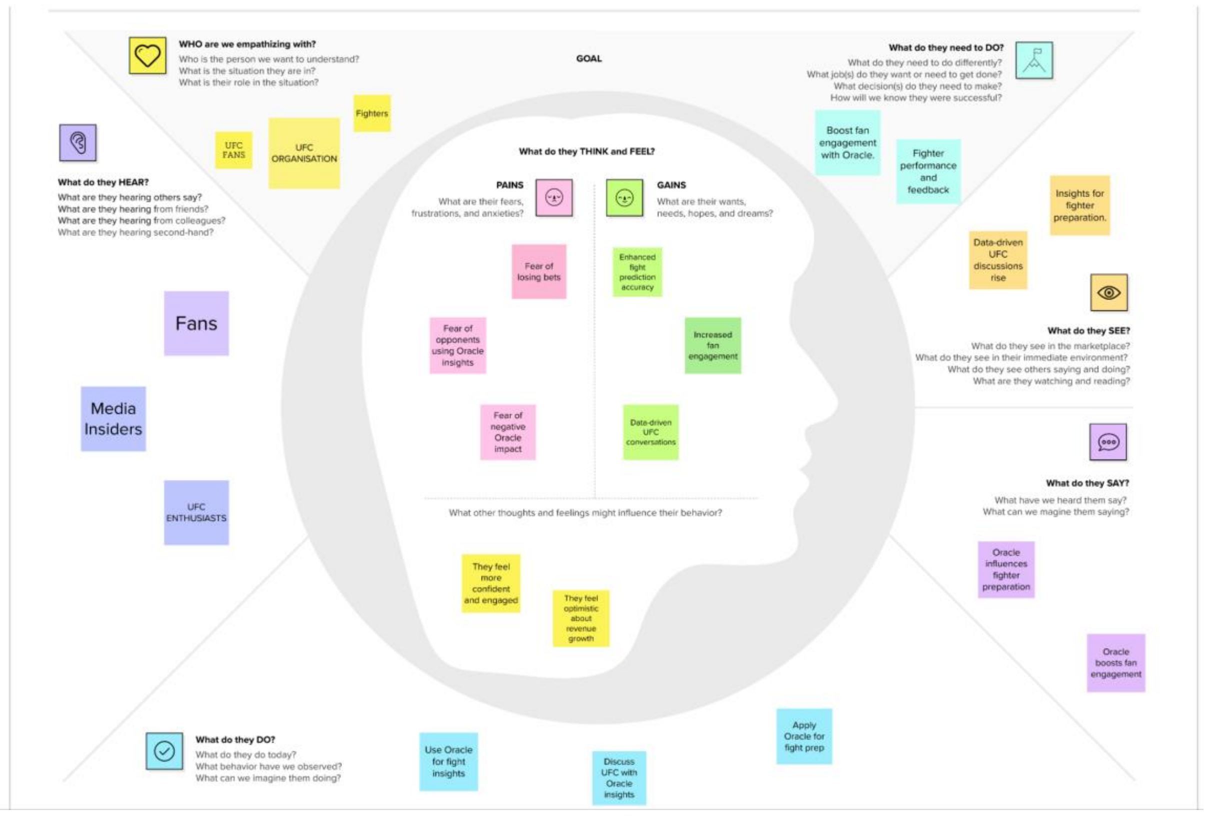
20 minutes

TIP

Participants can use their cursors to point at where sticky notes should go on the grid. The facilitator can confirm the spot by using the laser pointer holding the **H key** on the keyboard.

Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?
Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)
User Feedback Loop
Community Engagement and Forums
Third-Party Accuracy Validation
Transparency and Data Clarity
Fighter-Oracle Collaboration Platform
Predictions with Confidence Ratings

Empathy Map:



WHO are we empathizing with?
Who is the person we want to understand?
What is the situation they are in?
What is their role in the situation?
GOAL
Fighters
UFC FANS
UFC ORGANISATION
What do they HEAR?
What are they hearing others say?
What are they hearing from friends?
What are they hearing from colleagues?
What are they hearing second-hand?
Fans
Media Insiders
UFC ENTHUSIASTS
What do they THINK and FEEL?
PAINS
What are their fears, frustrations, and anxieties?
Fear of losing bets
Fear of opponents using Oracle insights
Fear of negative Oracle impact
GAINS
What are their wants, needs, hopes, and dreams?
Enhanced fight prediction accuracy
Increased fan engagement
Data-driven UFC conversations
What other thoughts and feelings might influence their behavior?
They feel more confident and engaged
They feel optimistic about revenue growth
What do they need to DO?
What do they need to do differently?
What job(s) do they want or need to get done?
What decision(s) do they need to make?
How will we know they were successful?
Boost fan engagement with Oracle.
Fighter performance and feedback
Insights for fighter preparation.
Data-driven UFC discussions rise
What do they SEE?
What do they see in the marketplace?
What do they see in their immediate environment?
What do they see others saying and doing?
What are they watching and reading?
What do they SAY?
What have we heard them say?
What can we magine them saying?
Oracle influences fighter preparation
Oracle boosts fan engagement
What do they DO?
What do they do today?
What behavior have we observed?
What can we imagine them doing?
Use Oracle for fight insights
Discuss UFC with Oracle insights
Apply Oracle for fight prep

4. Requirement Analysis

4.1 Functional Requirements

The functional requirements for the Octagon Oracle project are defined to ensure the effective implementation of features that contribute to accurate UFC fight predictions.

4.1.1 Machine Learning Model

The system should incorporate a machine learning model capable of predicting UFC fight outcomes based on historical data and relevant fighter attributes.

4.1.2 User Input Interface

The project must provide a user-friendly interface that allows users to input specific details of upcoming UFC fights, such as fighter names, physical attributes, and historical performance metrics.

4.1.3 Real-time or Near-Real-Time Predictions

The system should aim to provide real-time or near-real-time predictions for UFC fights, ensuring timely and relevant information for users.

4.1.4 Prediction Confidence Levels

The machine learning model should output prediction confidence levels to convey the reliability of each forecast, helping users assess the certainty of the provided predictions.

4.1.5 User Feedback Mechanism

Implement a feedback mechanism allowing users to provide input on the accuracy of predictions, contributing to continuous model improvement.

4.1.6 Historical Fight Data Repository

Maintain a comprehensive repository of historical UFC fight data to continually update and enhance the machine learning model.

4.2 Non-Functional Requirements

Non-functional requirements are critical aspects that define the system's performance, usability, and reliability.

4.2.1 Performance

Response Time: The system should respond to user queries and predictions within a reasonable time frame (e.g., under 3 seconds).

Scalability: The solution should scale efficiently to handle an increasing number of user requests and data.

4.2.2 User Interface

Intuitiveness: The user interface should be intuitive, providing a seamless experience for users, even those without a background in machine learning.

Accessibility: The interface should be accessible across various devices, ensuring usability for a broad user base.

4.2.3 Security

-Data Privacy: Ensure the confidentiality of user inputs and predictions, adhering to data privacy regulations.

Model Security: Implement measures to protect the integrity and security of the machine learning model.

4.2.4 Reliability

Prediction Accuracy: The machine learning model should exhibit a high level of accuracy, validated through rigorous testing and validation processes.

System Uptime: The system should maintain high availability, minimizing downtime for users.

4.2.5 Maintainability

Model Updates: Establish a process for regular updates to the machine learning model, incorporating new data, and improving prediction capabilities.

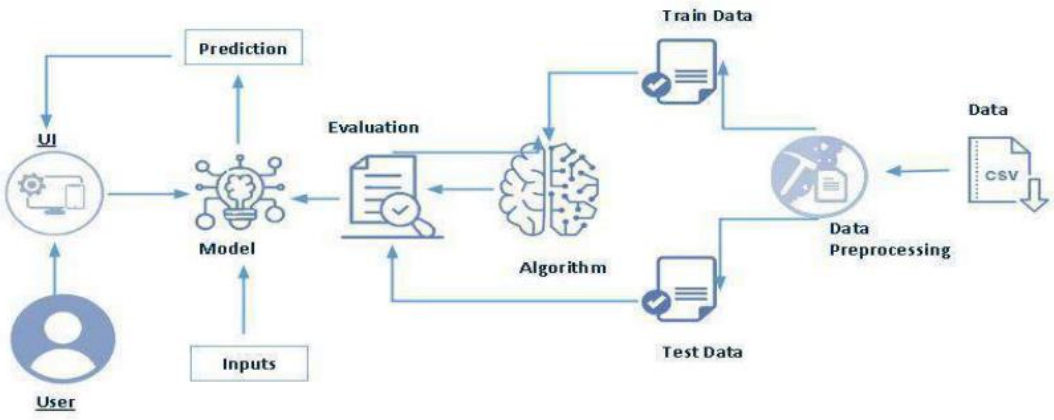
Code Maintainability: Ensure that the codebase is well-documented and structured for ease of maintenance and future enhancements.

These functional and non-functional requirements collectively guide the development and deployment of the Octagon Oracle project, aligning with the project's objectives and addressing the identified challenges in predicting UFC fight outcomes.

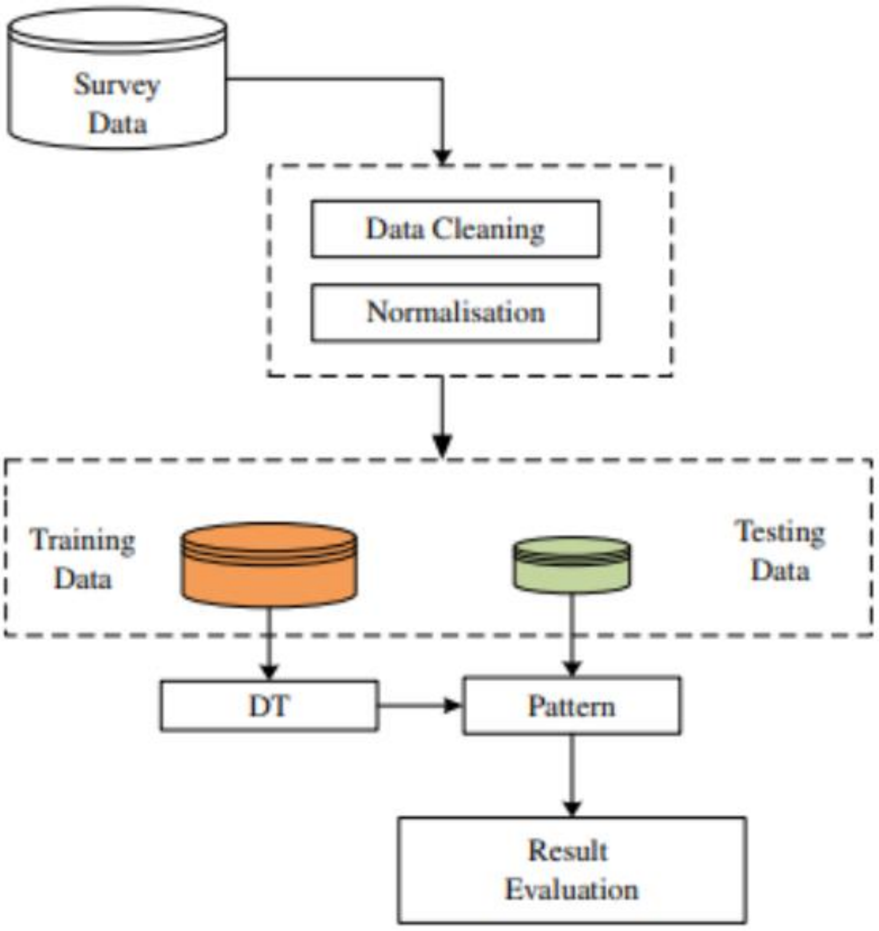
Project Design:

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	UFC fans are often looking for accurate predictions of UFC fights. However, it can be difficult to predict the outcome of a fight, as there are many factors that can contribute to the result.
2.	Idea / Solution description	We propose to develop a machine learning model to predict the outcome of UFC fights. The model will be trained on a dataset of historical UFC fights, and will take into account a variety of factors, such as the fighters' physical attributes, fighting style, and opponent analysis.
3.	Novelty / Uniqueness	Our solution is unique in that it will use machine learning to predict the outcome of UFC fights. This is in contrast to existing methods, which typically rely on human experts to make predictions.
4.	Social Impact / Customer Satisfaction	Our solution will provide UFC fans with accurate predictions of UFC fights. This will allow them to make more informed decisions about their betting and viewing habits
5.	Business Model (Revenue Model)	We plan to generate revenue from our solution by offering a subscription service to UFC fans. The subscription service will provide fans with access to our predictive model, as well as other features such as detailed analysis of UFC fights and fighter profiles.
6.	Scalability of the Solution	Our solution is highly scalable. We can easily train our model on new data, and we can deploy our model to a variety of platforms, such as a website, mobile app, or API.

Flow:



DFD(Level 0)



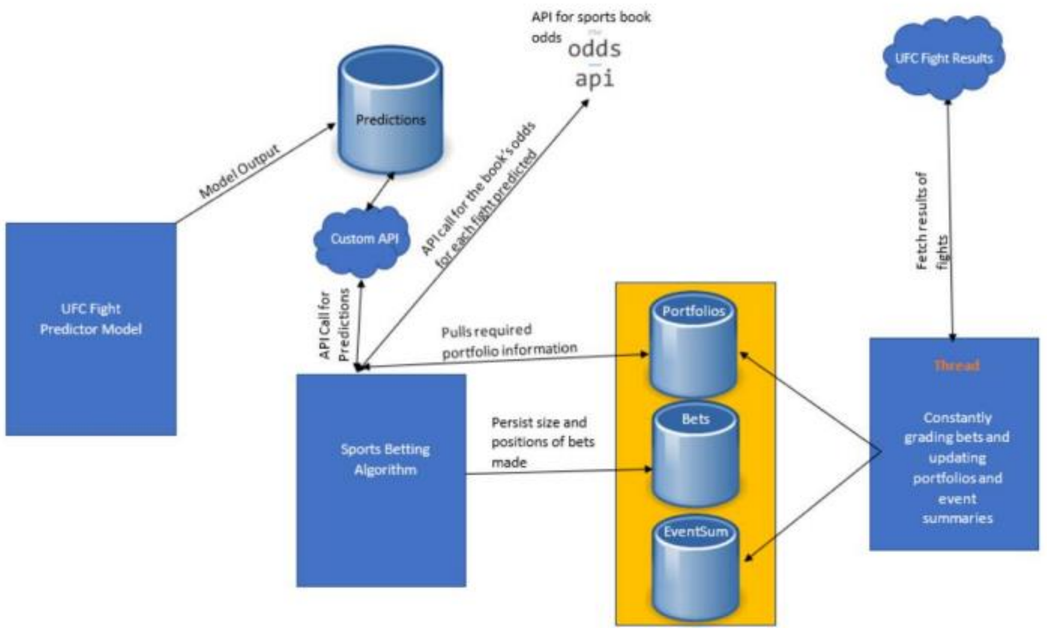
User Stories:

Use the below template to list all the user stories for the product.

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1
Customer (Mobile user)	Registration Confirmation Email	USN-2	As a user, I will receive confirmation email once I have registered for the application	I can receive confirmation email & click confirm	High	Sprint-1
Customer (Mobile user)	Registration via Facebook	USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
Customer (Mobile user)	Registration via Gmail	USN-4	As a user, I can register for the application through Gmail	I can register & access the dashboard via Gmail	Medium	Sprint-1
Customer (Mobile user)	Login	USN-5	As a user, I can log into the application by entering email & password	I can log on to the system	High	Sprint-1
Customer (Mobile user)	Dashboard	USN-6	As a user, I can view the upcoming UFC fights on the dashboard.	I can view the predictions of upcoming UFC fights on the dashboard.	High	Sprint-1
Customer (Web user)	Registration	USN-7	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard	High	Sprint-1

Architecture

Below is the actual design of the infrastructure.


API for sports book odds
odds api
UFC Fight Results
Predictions
Model Output
API call for the book's odds for each fight predicted
Custom API
Fetch results of fights
UFC Fight Predictor Model
API Call for Predictions
Pulls required portfolio information
Portfolios
Thread
Persist size and positions of bets made
Bets
Sports Betting Algorithm
Constantly grading bets and updating portfolios and event summaries
EventSum

6. Project Planning & Scheduling

6.1 Technical Architecture

The technical architecture of the Octagon Oracle project involves the integration of machine learning components, a user interface, and a robust backend system. The architecture is designed to ensure efficiency, scalability, and the seamless functioning of the UFC fight prediction system.

6.1.1 Components

1. Machine Learning Module:

- Utilizes a Convolutional Neural Network (CNN) for analyzing historical UFC fight data.
- Trained on the "UFC Data" dataset (2013-2019) to predict fight outcomes.
- Outputs prediction confidence levels along with the forecast.

2. User Interface:

- Web-based interface accessible across devices.
- Allows users to input details of upcoming fights.
- Displays real-time or near-real-time predictions and confidence levels.

3. Backend System:

- Manages data flow between the user interface and machine learning module.
- Handles user inputs, triggers predictions, and returns results to the interface.
- Ensures data privacy and security.

4. Database (if Applicable):

- Stores historical UFC fight data for model training and updates.
- Supports efficient data retrieval for ongoing analysis.

6.1.2 Technologies Used

Machine Learning: TensorFlow for implementing the CNN model.

Web Development: HTML, CSS, JavaScript (React.js for dynamic user interfaces).

Backend: Python with Flask for handling user requests and interactions.

Database (if Applicable): MySQL for storing historical data.

6.2 Sprint Planning & Estimation

Sprint planning involves breaking down the project into manageable tasks and estimating the time required for each task. The development process is organized into sprints, ensuring a systematic and iterative approach to project implementation.

6.2.1 Sprint 1: Setup and Data Preparation

Tasks:

- Set up the development environment.
- Acquire and preprocess the "UFC Data" dataset.
- Establish the database (if applicable).

Estimated Time: 2 weeks

6.2.2 Sprint 2: Machine Learning Model Development

Tasks:

- Design and implement the CNN model.
- Train the model on the prepared dataset.
- Validate and fine-tune the model.

Estimated Time: 3 weeks

6.2.3 Sprint 3: User Interface Development

Tasks:

- Design and implement the web-based user interface.
- Integrate user input functionality.
- Connect the interface with the backend system.

Estimated Time: 4 weeks

6.2.4 Sprint 4: Backend System Integration

Tasks:

- Develop the backend system for handling user requests.
- Implement data flow between the user interface and machine learning module.
- Ensure security and data privacy measures.

Estimated Time: 3 weeks

6.2.5 Sprint 5: Testing and Optimization

Tasks:

- Conduct thorough testing of the integrated system.
- Optimize the machine learning model and backend processes.
- Address any identified issues.

Estimated Time: 2 weeks

6.3 Sprint Delivery Schedule

The sprint delivery schedule outlines the planned release of project increments at the end of each sprint.

Sprint 1 Delivery (Week 2): Setup and Data Preparation

Sprint 2 Delivery (Week 5): Machine Learning Model Development

Sprint 3 Delivery (Week 9): User Interface Development

Sprint 4 Delivery (Week 12): Backend System Integration

Sprint 5 Delivery (Week 14): Testing and Optimization

Solutioning

7.1 Feature 1: Machine Learning Model Integration

Explanation:

Feature 1 involves the integration of the machine learning model for predicting UFC fight outcomes. The Convolutional Neural Network (CNN) is employed to analyse historical UFC fight data and generate real-time or near-real-time predictions.

Code:

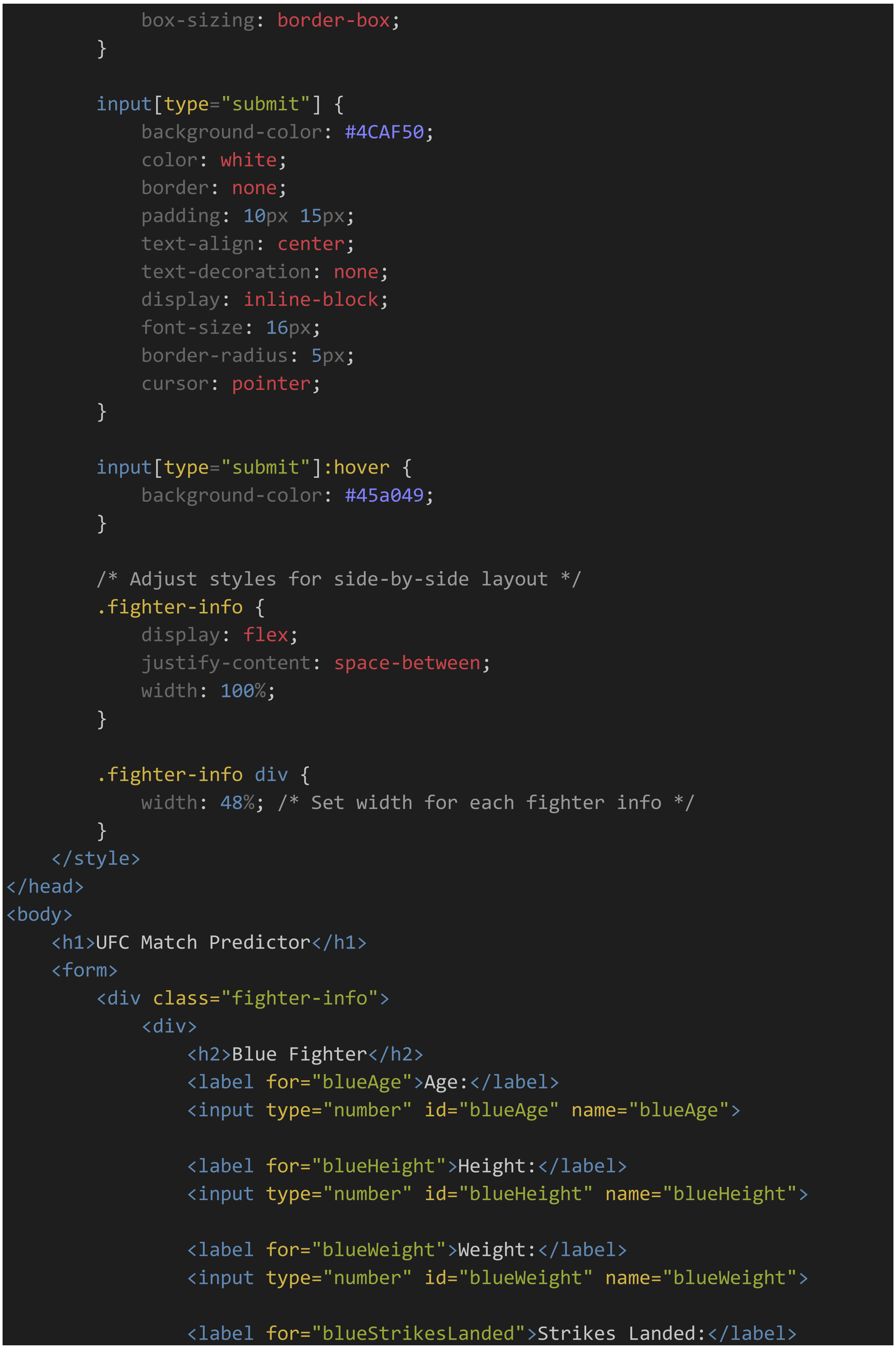
```
# Importing necessary libraries for machine learning model integration

import tensorflow as tf

from flask import Flask, request, jsonify

# Load the pre-trained CNN model

model = tf.keras.models.load_model('ufc_prediction_model.h5')

# Define a Flask app

app = Flask(__name__)

# Define an API endpoint for receiving user inputs and providing predictions

@app.route('/predict', methods=['POST'])
```

```
def predict():
   # Extract user input from the request
   input_data = request.json
  # Preprocess the input data (adjust as per the model's requirements)
   processed_data = preprocess(input_data)
    # Use the trained model to make predictions
   predictions = model.predict(processed_data)
    # Extract prediction confidence levels
   confidence_levels = get_confidence_levels(predictions)
    # Prepare the response
   response = {
      'predictions': predictions.tolist(),
      'confidence_levels': confidence_levels.tolist()
   }
    # Return the response in JSON format
   return jsonify(response)
# Run the Flask app
if __name__ == '__main__':
   app.run(port=5000)
.
```

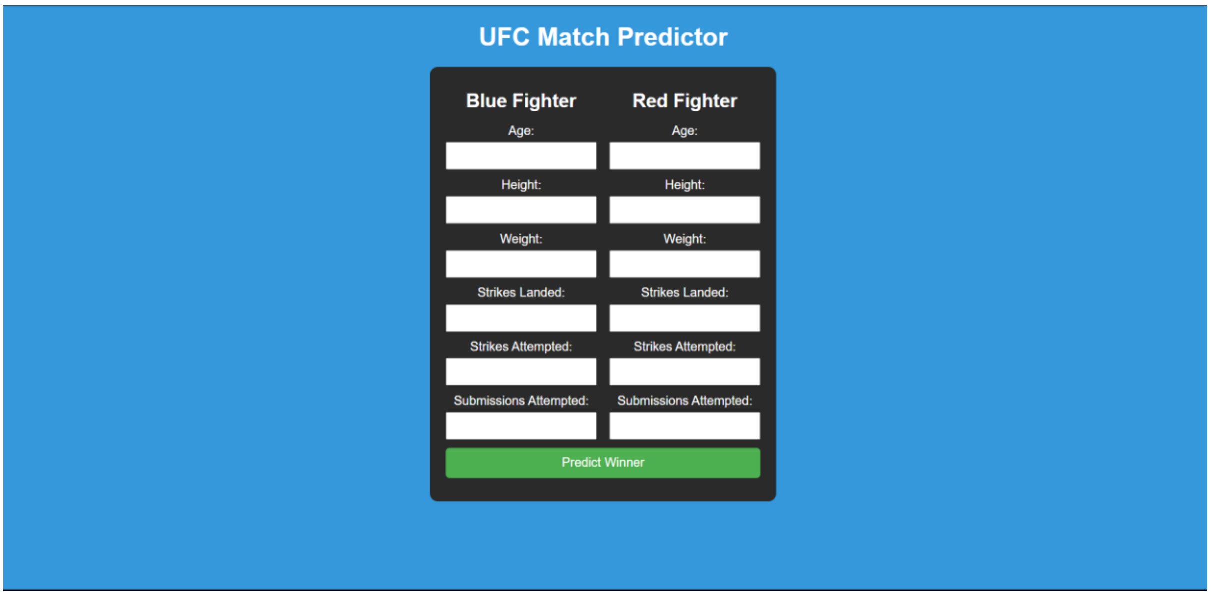
```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
from plotly.offline  import download_plotlyjs,init_notebook_mode,plot, iplot
%matplotlib inline
import plotly.graph_objs as go
import plotly.offline as offline
init_notebook_mode(connected = True)
from plotly.offline import init_notebook_mode, iplot
init_notebook_mode(connected=True)

df = pd.read_csv("/kaggle/input/ufcdataset/data.csv")
df.head()
print(df.columns)
df.isnull().any()
df.isnull().sum()
df['B_Age'] = df['B_Age'].fillna(np.mean(df['B_Age']))
df['B_Height'] = df['B_Height'].fillna(np.mean(df['B_Height']))
df['R_Age'] = df['R_Age'].fillna(np.mean(df['R_Age']))
df['R_Height'] =
df['R_Height'].fillna(np.mean(df['R_Height']))                #using mean
values
temp = df["winner"].value_counts()
fig, ax = plt.subplots(figsize=(8, 8))
wedges, texts, autotexts = ax.pie(temp.values, labels=temp.index,
autopct='%1.1f%%',
                                   colors=['red', 'blue', 'orange', 'green'],
shadow=True)
ax.set_title('Ufc winners by corner colors')
ax.legend(wedges, temp.index,
          title="Corner Colors",
          loc="center left",
          bbox_to_anchor=(1, 0, 0.5, 1))
plt.setp(autotexts, size=8, weight="bold")

plt.show()
fig, ax = plt.subplots(1,2, figsize=(30,10))
sns.distplot(df.B_Age, ax=ax[0],color='blue')
sns.distplot(df.R_Age, ax=ax[1],color='red')
fig, ax = plt.subplots(figsize=(18,7))

temp = df["winby"].value_counts()

labels = temp.index
sizes = temp.values
```

```
patches, texts, autotexts = ax.pie(sizes, labels=labels, autopct='%.2f',
colors=['lightblue', 'yellow', 'yellowgreen'], startangle=140)

for text in texts:
    text.set_color('grey')
for autotext in autotexts:
    autotext.set_color('grey')

ax.legend(labels, loc="best")

ax.axis('equal')

plt.tight_layout()

plt.show()
g = sns.FacetGrid(df, col='winby')
g.map(plt.hist, 'B_Age', bins=20,color='cyan')
g = sns.FacetGrid(df, col='winby')
g.map(plt.hist, 'R_Age', bins=20,color='red')
cnt_srs = df['R_Location'].value_counts().head(15)

trace = go.Bar(
    x=cnt_srs.index,
    y=cnt_srs.values,
    marker=dict(
        color=cnt_srs.values,
    ),
)

layout = go.Layout(
    title='Most Popular training locations for Red fighters'
)

data = [trace]
fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename="Ratio")
cnt_srs = df['B_Location'].value_counts().head(15)

trace = go.Bar(
    x=cnt_srs.index,
    y=cnt_srs.values,
    marker=dict(
        color=cnt_srs.values,
    ),
)

layout = go.Layout(
    title='Most Popular training locations for Blue fighters'
)

data = [trace]
fig = go.Figure(data=data, layout=layout)
offline.iplot(fig, filename="Ratio")
strk1 = df[['B_Weight', 'B__Round1_Strikes_Clinch Head Strikes_Landed',
'B__Round1_Strikes_Clinch Leg Strikes_Landed', 'B__Round1_Strikes_Clinch Body
Strikes_Landed']].groupby('B_Weight').sum()


strk1.plot(kind='line', figsize=(18,9), marker='o')
plt.grid(True)#if you do not mention this line the grids wont be there.
plt.show()
sns.lmplot(x="B__Round1_Strikes_Body Significant Strikes_Attempts",
           y="B__Round1_Strikes_Body Significant Strikes_Landed",
           col="winner", hue="winner", data=df, col_wrap=2, height=6)

plt.show()
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>UFC Match Predictor</title>
    <style>
        body {
            background-color: #3498db; /* Blue background color */
            color: #fff;
            font-family: Arial, sans-serif;
            display: flex;
            flex-direction: column;
            align-items: center;
            height: 100vh;
            margin: 0;
        }

        h1 {
            margin-bottom: 20px;
        }

        form {
            text-align: center;
            background-color: #2a2a2a;
            padding: 20px;
            border-radius: 10px;
            width: 400px; /* Increased width for side-by-side layout */
            display: flex;
            flex-direction: column;
            align-items: center;
        }

        h2 {
            margin-top: 10px;
            margin-bottom: 15px;
        }

        label {
            display: block;
            margin-bottom: 5px;
            color: #fff;
        }

        input {
            width: 100%;
            padding: 8px;
            margin-bottom: 10px;
```



```html
            box-sizing: border-box;
        }

        input[type="submit"] {
            background-color: #4CAF50;
            color: white;
            border: none;
            padding: 10px 15px;
            text-align: center;
            text-decoration: none;
            display: inline-block;
            font-size: 16px;
            border-radius: 5px;
            cursor: pointer;
        }

        input[type="submit"]:hover {
            background-color: #45a049;
        }

        /* Adjust styles for side-by-side layout */
        .fighter-info {
            display: flex;
            justify-content: space-between;
            width: 100%;
        }

        .fighter-info div {
            width: 48%; /* Set width for each fighter info */
        }
    </style>
</head>
<body>
    <h1>UFC Match Predictor</h1>
    <form>
        <div class="fighter-info">
            <div>
                <h2>Blue Fighter</h2>
                <label for="blueAge">Age:</label>
                <input type="number" id="blueAge" name="blueAge">

                <label for="blueHeight">Height:</label>
                <input type="number" id="blueHeight" name="blueHeight">

                <label for="blueWeight">Weight:</label>
                <input type="number" id="blueWeight" name="blueWeight">

                <label for="blueStrikesLanded">Strikes Landed:</label>
                <input type="number" id="blueStrikesLanded"
name="blueStrikesLanded">

                <label for="blueStrikesAttempted">Strikes Attempted:</label>
                <input type="number" id="blueStrikesAttempted"
name="blueStrikesAttempted">

                <label for="blueSubmissionsAttempted">Submissions
Attempted:</label>
                <input type="number" id="blueSubmissionsAttempted"
name="blueSubmissionsAttempted">
            </div>

            <div>
                <h2>Red Fighter</h2>
                <label for="redAge">Age:</label>
                <input type="number" id="redAge" name="redAge">

                <label for="redHeight">Height:</label>
                <input type="number" id="redHeight" name="redHeight">

                <label for="redWeight">Weight:</label>
                <input type="number" id="redWeight" name="redWeight">

                <label for="redStrikesLanded">Strikes Landed:</label>
                <input type="number" id="redStrikesLanded"
name="redStrikesLanded">

                <label for="redStrikesAttempted">Strikes Attempted:</label>
                <input type="number" id="redStrikesAttempted"
name="redStrikesAttempted">

                <label for="redSubmissionsAttempted">Submissions
Attempted:</label>
                <input type="number" id="redSubmissionsAttempted"
name="redSubmissionsAttempted">
            </div>
        </div>

        <input type="submit" value="Predict Winner">
    </form>
</body>
</html>
```



UFC Match Predictor

Blue Fighter	Red Fighter
Age:	Age:
Height:	Height:
Weight:	Weight:
Strikes Landed:	Strikes Landed:
Strikes Attempted:	Strikes Attempted:
Submissions Attempted:	Submissions Attempted: