# EquiSync™

EquiSync™ is an AI-powered performance tracking system for racehorses. This system utilizes deep learning and computer vision techniques to analyze biomechanical and physiological data, predicting a horse's performance score.

## 📂 **File Structure:-**

- `app.py` - Flask application for making predictions using the trained LSTM model.

- `training.py` - Script for training the LSTM model using horse biomechanics, vital signs, and race performance data.

- `Horse\_Biomechanics.csv` - Dataset containing biomechanical measurements of horses.

- `Horse\_VitalSigns.csv` - Dataset containing vital sign records of horses.

- `Race\_Performance.csv` - Dataset containing race performance metrics of horses.

⚙️ **Installation:-**

1. Clone this repository:

```bash

git clone https://github.com/your-repo/equisync.git

cd equisync

**Install required packages**:-

pip install -r requirements.txt

**Train the model**(if not trained):-

python training.py

**Run the application**:-

python app.py

### The application will run on <http://127.0.0.1:5000>.

**API Endpoints**

**1. Predict Performance Score**

* **Endpoint:** /predict
* **Method:** POST
* **Description:** Predicts the horse's performance score based on its biomechanics and vital signs.

**Request Format**

json

{

"Stride\_Length": 2.5,

"Acceleration": 3.2,

"Speed": 60.5,

"Heart\_Rate": 120,

"Oxygen\_Level": 98

}

**Response Format**

json

{

"Performance\_Score": 85.7

}

**Testing the API in Postman**

1. Select **POST** method.
2. Set the URL to http://127.0.0.1:5000/predict.
3. Go to **Body** → **raw** → **JSON** format and enter the request payload.
4. Click **Send** and receive the predicted performance score.

**🛠 Dependencies**

* Flask
* Flask-CORS
* TensorFlow
* Scikit-learn
* Pandas
* NumPy

**Notes:-**

* Ensure that Horse\_Biomechanics.csv, Horse\_VitalSigns.csv, and Race\_Performance.csv are in the same directory.
* The input JSON must contain the features used in training.
* The model should be trained before running predictions.

---

**Requirements:-**

Flask==3.0.3

Flask-Cors==5.0.0

tensorflow==2.14.0

pandas==2.2.3

numpy==2.0.2

scikit-learn==1.5.2

**API Endpoint for Render Deployment**

Once deployed on Render, use:

[https://##your-app-name.onrender.com##/predict](NULL)

for making predictions in Postman.