# Running the PSO-Optimized Neural Network Project on Kaggle

## Overview

In this project, we aim to optimize a neural network using PSO to predict football match outcomes (Win, Draw, Loss) based on features extracted from the SQLite football database. The model's training performance and evaluation metrics are observed under multiple configurations including changes in:  
- Dataset size (e.g., 5000, 10000, 15000 samples)  
- Number of PSO generations (e.g., 20, 30)  
- Informant counts (3, 4, 5)  
  
Kaggle provides an efficient runtime platform for large-scale experimentation.

## Step-by-Step Instructions

### Step 1: Sign in and Launch a New Notebook

1. Go to https://www.kaggle.com/  
2. Sign in using your credentials.  
3. Click on your profile icon → “Your Work” → “Code” → “+ New Notebook”.  
4. Name the notebook (e.g., pso\_football\_nn\_kaggle\_run).  
5. Choose CPU or GPU (GPU is optional but useful for faster NN training).  
6. Set the notebook language to Python and the environment to Notebook.

### Step 2: Upload or Attach Dataset

Option 1: Add Public Dataset (Soccer by Hugo Mathien):

- Click on the folder icon in the right sidebar.  
- Click “+ Add Data”.  
- Search for “Soccer” by hugomathien.  
- Click “Add” to attach the dataset.

Option 2: Upload Your Own SQLite File:

- Click the “Upload” tab in “Add Data”.  
- Upload your .sqlite file.  
- It will appear under /kaggle/input/your-dataset-folder/.

Update SQLite path in your code:

conn = sqlite3.connect('/kaggle/input/your-dataset-folder/database.sqlite')

### Step 3: Prepare the Notebook

Option A – Paste Code in Cells:

- Divide your complete code into logical sections.  
- Paste each section in a new cell.

Option B – Upload Python Script:

- Upload it via “+ Add Data” or drag into /kaggle/working/.  
- Run it with: %run /kaggle/working/your\_script.py

### Step 4: Update Dataset Path and Parameters

- Set correct SQLite path.  
- Modify SQL LIMIT clause as needed.  
- Adjust PSO parameters like GENERATIONS and SWARM\_SIZE.

### Step 5: Run and Monitor Outputs

1. Use “Run All” or execute cells individually.  
2. Monitor accuracy logs, confusion matrices, classification reports, and runtime.

### Step 6: Save or Export Results

- Click “Save Version” to preserve work.  
- Download graphs using right-click or plt.savefig().  
- Save CSV summaries or model files to /kaggle/working/ for download.

## Performance Considerations

• Runtime: Kaggle allows 12-hour CPU runtime.  
• Model Size: Suitable for up to 15000 rows.  
• File Upload: Larger SQLite files can be reused.  
• Speed: Faster than Colab or local machines for PSO workloads.  
• Copy/Paste: Uploading scripts is more efficient than pasting large code blocks.

## Tips for Beginners

- Restart the kernel if the session times out.  
- Avoid printing full model summaries in loops.  
- Test on smaller datasets before full runs.