import streamlit as st

import pandas as pd

# ✅ Full list of 17 NRL teams

all\_teams = [

'Broncos', 'Raiders', 'Bulldogs', 'Sharks', 'Titans', 'Sea Eagles', 'Storm',

'Knights', 'Cowboys', 'Eels', 'Panthers', 'Rabbitohs', 'Dragons', 'Roosters',

'Warriors', 'Tigers', 'Dolphins'

]

# ✅ Sample match history (you can expand later)

data = {

'Home Team': ['Storm', 'Panthers', 'Roosters', 'Storm', 'Broncos', 'Panthers'],

'Away Team': ['Broncos', 'Storm', 'Panthers', 'Roosters', 'Panthers', 'Roosters'],

'Winner': ['Storm', 'Panthers', 'Roosters', 'Storm', 'Panthers', 'Panthers']

}

df = pd.DataFrame(data)

# 🌟 Streamlit App

st.title("🏉 NRL Match Winner Predictor")

home\_team = st.selectbox("Select Home Team", all\_teams)

away\_team = st.selectbox("Select Away Team", [team for team in all\_teams if team != home\_team])

def predict\_winner(home, away):

# Check if there's match history

home\_wins = df[(df['Home Team'] == home) & (df['Winner'] == home)].shape[0]

away\_wins = df[(df['Away Team'] == away) & (df['Winner'] == away)].shape[0]

# Placeholder for expert analysis results (to be added later)

expert\_tips = None # You'll replace this with real scraped/extracted data

team\_with\_most\_tips = home # or away depending on what scraping finds

summary = "Most analysts favor " + team\_with\_most\_tips # Replace with actual summary

if expert\_tips: # when expert data is available

winner = team\_with\_most\_tips

reason = summary

return f"🤖 Expert Prediction: \*\*{winner}\*\*\n🧠 Reason: {reason}"

elif home\_wins > away\_wins:

return f"{home} (based on stronger home history)"

elif away\_wins > home\_wins:

return f"{away} (based on stronger away form)"

else:

return "📉 No strong data available — matchup is balanced or not recorded."

# Button logic

if st.button("Predict Winner"):

result = predict\_winner(home\_team, away\_team)

st