**Advanced Future Engineering and Sentiment Analysis:**

**1. Advanced Performance Trend Features**

import pandas as pd

import numpy as np

# Example performance dataset

performance\_df = pd.read\_csv("player\_performance.csv")

# Rolling average stats (last 5 games, last 10 games)

performance\_df["goals\_5game\_avg"] = performance\_df.groupby("player")["goals"].transform(lambda x: x.rolling(5, min\_periods=1).mean())

performance\_df["assists\_5game\_avg"] = performance\_df.groupby("player")["assists"].transform(lambda x: x.rolling(5, min\_periods=1).mean())

performance\_df["goals\_10game\_avg"] = performance\_df.groupby("player")["goals"].transform(lambda x: x.rolling(10, min\_periods=1).mean())performance\_df["assists\_10game\_avg"] = performance\_df.groupby("player")["assists"].transform(lambda x: x.rolling(10, min\_periods=1).mean())

# Performance consistency (variance of ratings in last 5 games)

performance\_df["rating\_variability"] = performance\_df.groupby("player")["rating"].transform(lambda x: x.rolling(5, min\_periods=1).std())

print(performance\_df.head())

**2. Injury & Availability Features**

# Injury dataset

injury\_df = pd.read\_csv("injury\_data.csv")

# Total missed matches

injury\_df["matches\_missed"] = injury\_df["injury\_days"] // 7

# Injury frequency

injury\_df["injury\_count"] = injury\_df.groupby("player")["injury\_type"].transform("count")

# Injury severity score (days missed per season)

injury\_df["injury\_severity"] = injury\_df["injury\_days"] / injury\_df["seasons\_played"]

print(injury\_df.head())

**3. Contract-Related Features**

# Contract dataset

contracts\_df = pd.read\_csv("contract\_data.csv")

# Contract duration remaining in months

contracts\_df["contract\_remaining\_months"] = (pd.to\_datetime(contracts\_df["contract\_end"]) - pd.to\_datetime("today")).dt.days // 30

# Player loyalty score (years at current club)

contracts\_df["club\_loyalty\_years"] = (pd.to\_datetime("today") - pd.to\_datetime(contracts\_df["joined\_date"])).dt.days // 365

print(contracts\_df.head())

**4. Sentiment Analysis from Social Media**

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer

# Example social media dataset

tweets\_df = pd.read\_csv("twitter\_sentiment.csv")

analyzer = SentimentIntensityAnalyzer()

tweets\_df["sentiment"] = tweets\_df["tweet"].apply(lambda x: analyzer.polarity\_scores(str(x))["compound"])

# Aggregate sentiment per player

sentiment\_features = tweets\_df.groupby("player")["sentiment"].agg(["mean", "std", "count"]).reset\_index()

sentiment\_features.rename(columns={"mean":"sentiment\_mean", "std":"sentiment\_volatility", "count":"mention\_count"}, inplace=True)

print(sentiment\_features.head())

**5. Merging All Feature Sets**

# Merge performance, injury, contracts, and sentiment

final\_features = performance\_df.merge(injury\_df[["player","injury\_count","injury\_severity"]], on="player", how="left")

final\_features = final\_features.merge(contracts\_df[["player","contract\_remaining\_months","club\_loyalty\_years"]], on="player", how="left")

final\_features = final\_features.merge(sentiment\_features, on="player", how="left")

# Fill missing values

final\_features = final\_features.fillna(0)

# Save final dataset

final\_features.to\_csv("week3\_final\_feature\_set.csv", index=False)

print("Final feature set saved: week3\_final\_feature\_set.csv")

**6. Visualization for Reports**

import matplotlib.pyplot as plt

# Sentiment trend plot for a single player

player\_name = "Lionel Messi"

player\_tweets = tweets\_df[tweets\_df["player"] == player\_name]

player\_tweets.groupby("date")["sentiment"].mean().plot(kind="line", title=f"Sentiment Trend for {player\_name}")

plt.xlabel("Date")

plt.ylabel("Sentiment Score")

plt.show()

# Correlation heatmap between features

import seaborn as sns

corr = final\_features.corr(numeric\_only=True)

plt.figure(figsize=(10,6))

sns.heatmap(corr, cmap="coolwarm", annot=False)

plt.title("Feature Correlation Heatmap")

plt.show()