Joe Stella

Progress Report

BatOracle

Written Code(Unfinished):

import pandas as pd

import xgboost as xgb

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.metrics import mean\_squared\_error

from pybaseball import statcast

import warnings

# Suppress FutureWarning from pybaseball

warnings.simplefilter(action='ignore', category=FutureWarning)

batter\_stats = pd.read\_csv("batterstats.csv")

pitcher\_stats = pd.read\_csv("pitcherstats.csv")

# Retrieve the 2023 Statcast data

start\_date = "2023-01-01"

end\_date = "2023-12-31"

statcast\_data = statcast(start\_dt=start\_date, end\_dt=end\_date)

# Define features (batter and pitcher stats) and target variable (batter performance)

X = batter\_stats["woba"],pitcher\_stats["woba"]  # Features

y = statcast\_data["events"]  # Target variable

# Split data into train and test sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Standardize features

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

# Train an XGBoost model

xgb\_model = xgb.XGBRegressor()

xgb\_model.fit(X\_train\_scaled, y\_train)

# Make predictions

y\_pred = xgb\_model.predict(X\_test\_scaled)

# Evaluate the model

mse = mean\_squared\_error(y\_test, y\_pred)

print(f"Mean Squared Error: {mse:.2f}")

# Save the model

xgb\_model.save\_model("batter\_performance\_xgb\_model.xgb")

print("XGBoost model trained and saved successfully.")

**Data Collection Summary** In my pursuit to leverage machine learning for predicting baseball events, I have meticulously crafted a codebase that not only interfaces with the pybaseball library but also integrates diverse datasets to form a robust analytical foundation. The Statcast data for the year 2023, which I have fetched, is a treasure trove of information, capturing the intricate details of baseball events. Alongside this, I have collated historical batter and pitcher performance metrics from batterstats.csv and pitcherstats.csv. My code ensures a seamless data preprocessing workflow, including the suppression of warnings, and adeptly prepares the datasets for the subsequent model training phase. The datasets have been split into training and test sets with a strategic 80-20 division, followed by standardization to facilitate the training of the XGBoost model, a decision-tree-based ensemble machine learning algorithm that uses a gradient boosting framework.

**Challenges in Data Integration and Feature Selection** The task of aligning player identifiers across the datasets I’ve collected and the Statcast data has proven to be a formidable challenge. This misalignment of player IDs creates a significant barrier to achieving a cohesive dataset, which is crucial for the accurate prediction of baseball events. Furthermore, the critical process of feature selection from the extensive pool of potential data points is underway. Identifying the most impactful features that accurately reflect the complexities of player performance is vital to the success of the predictive model. To address this, I am delving into various feature selection methodologies to distill the dataset down to the most predictive elements, thereby enhancing the interpretability and precision of the model on new, unseen data.

**Plans for Upcoming Week** In the following week, I will be expanding the dataset to include additional years of data, thereby enriching the model’s learning capacity with a more comprehensive historical context. This expansion is expected to provide a deeper insight into player performance trends over time. Concurrently, I will be meticulously evaluating and finalizing the selection of features for the predictive model. This process will involve a careful examination of each potential feature’s contribution to the model’s accuracy. Throughout the week, I will be continuously monitoring the model’s performance metrics, with a keen focus on the mean squared error, to gauge the accuracy and effectiveness of the model as these enhancements are implemented.