

Data Visualization Website

This website focuses on the data visualization of football data, encompassing actual match results, generated predictions by AWS SageMaker, and sentiment analysis based on news data for five football teams.

The football teams include Arsenal, Chelsea, Liverpool, Manchester City and Manchester United.

Website Features and Technology

Public S3 link: <https://footballdatalandingpagebucket.s3.amazonaws.com/index.html>

Typescript: The code for uploading the numerical and text data was written in TypeScript. Football data is extracted from a CSV file, processed, and then stored in the DynamoDB database. Text data is extracted from News API, processed into sentiments, then stored in DynamoDB.

Numerical Data Display: Fetching the stored numerical data from DynamoDB, and then visualizing it with Plotly's JavaScript for insights into five football teams' stats. The data is split into actual and predicted sections. Predictions generated by AWS SageMaker's DeepAR algorithm. Both the actual data and predicted data are plotted together on the same line-chart graph but separated by distinct traces for easy comparison.

Sentiment Analysis Display: The sentiment analysis of each team is visualized using pie charts. They are generated through processed text data from the News DynamoDB table. This analysis utilizes an API from text-processing.com to create sentiments, offering a clear visual representation of public sentiment towards each team.

WebSockets: When the page loads, the client requests data from the server using WebSockets. The server then sends the data back through the same channel. If cloud data changes, the server broadcasts these updates to all connected clients via WebSockets, ensuring real-time data synchronization.

Serverless: The website makes use of AWS serverless features by handling backend tasks without the need to manage servers through Lambda functions. Lambda functions are automatically triggered by changes in specific DynamoDB tables.

Cloud-Based Infrastructure: All project data is stored in the cloud, leveraging AWS S3 serverless technology for hosting the website.

Machine Learning: Using Deep AR algorithm with AWS Sagemaker to train and test five teams' football data and synthetic data to help predict future match outcomes. Created training jobs for each team and synthetic data, created models for them, then used created endpoints to store the predictions of each team in DynamoDB.

Front-end: Frontend was created using pure HTML and JavaScript, while styling was done using Tailwind CSS.

Lambda Functions

processSentiments: This lambda function is triggered after there is a change in the News DynamoDB table. It uses text processing.com API to convert text data into sentiments. It then saves the sentiment in a new table called Sentiments.

wsConnect: This lambda function manages the websocket connection events.

wsDisconnect: This lambda function manages the websocket disconnection events.

wsDefault: This lambda function is the default websocket connection events.

wsMessage: This folder contains three files consisting of a websocket.mjs file, database.mjs file, and index.mjs file. It leverages lambda, an API Gateway, and DynamoDB to create that serverless system for real-time communication via WebSockets. It processes incoming WebSocket messages, queries and manipulates data in DynamoDB based on these messages and sends responses back to clients. The setup includes functionality for sending messages to specific clients, handling disconnects by cleaning up connection IDs in DynamoDB, and fetching as well as processing team-based data and sentiments for response messages.

plotSyntheticDataFromEndpoint: This lambda function uses AWS SageMaker to invoke a machine learning model for synthetic data predictions. It retrieves original data via an Axios request, uses SageMaker to retrieve the predicted data, then visualizes both sets of data using Plotly.

broadcastForSentiment: This lambda function uses websockets to broadcast new sentiment data added to the Sentiments DynamoDB table.

broadcastConnectedIDs: This lambda function uses websockets to broadcast new numerical data added to the FootballMatches DynamoDB table.

DynamoDB Tables

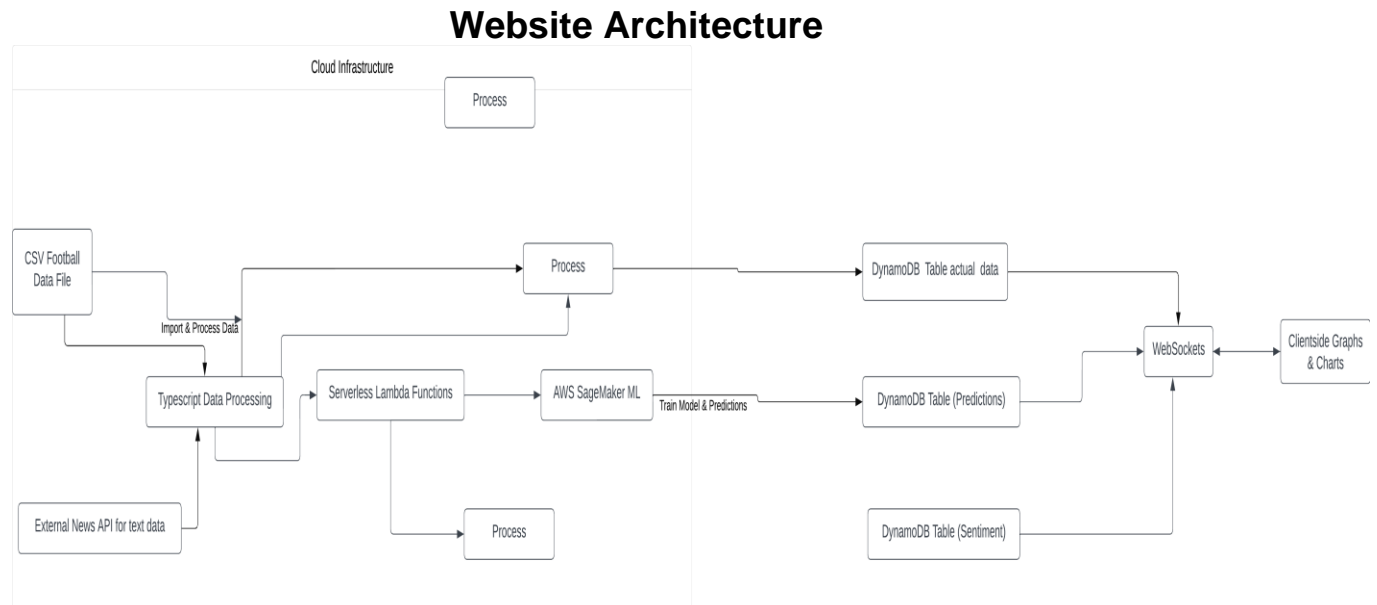
FootballMatches: Stores all numerical data for each team extracted from the CSV file.

FootballNews: Stores all text data for each team, gotten from News Api.

Sentiments: Stores the processed text data results as sentiments.

WebSocket: Stores all websocket connection IDs.

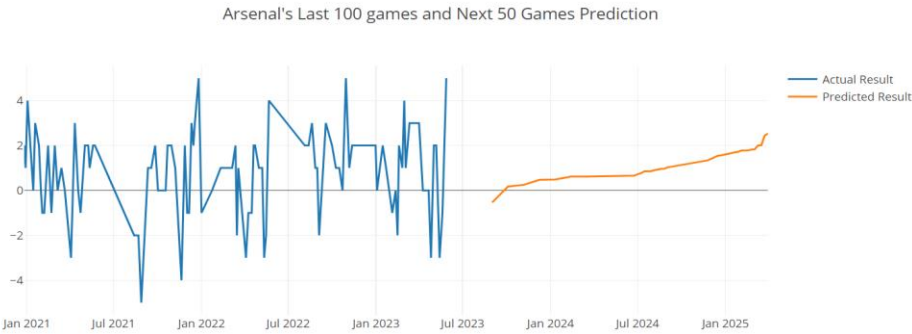
PredictedResults: Stores the predicted football results.



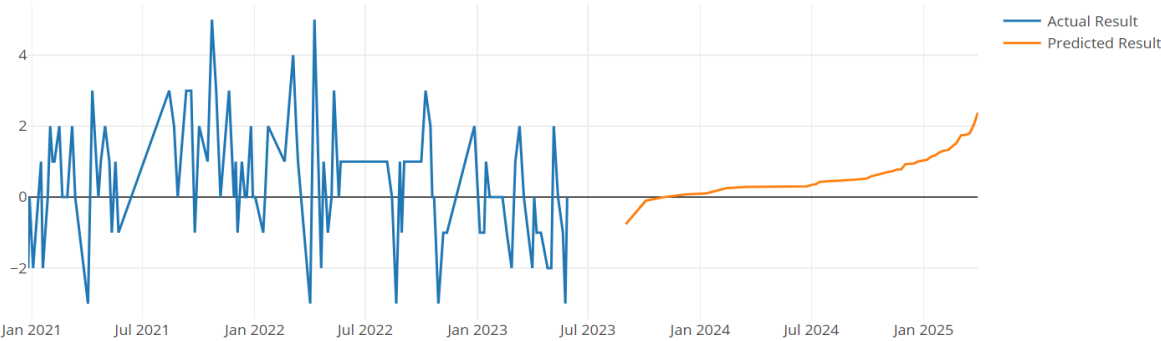
Website Screenshots



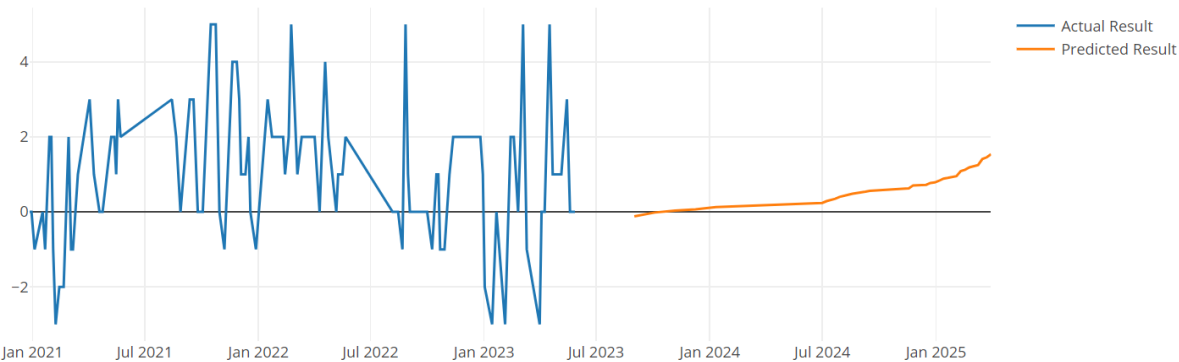
Actual and Predicted Data Visualisation



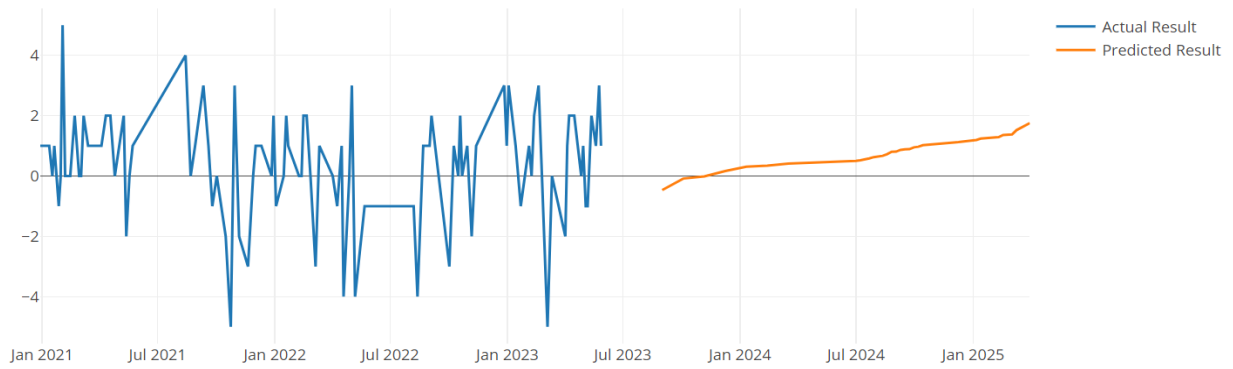
Chelsea's Last 100 games and Next 50 Games Prediction



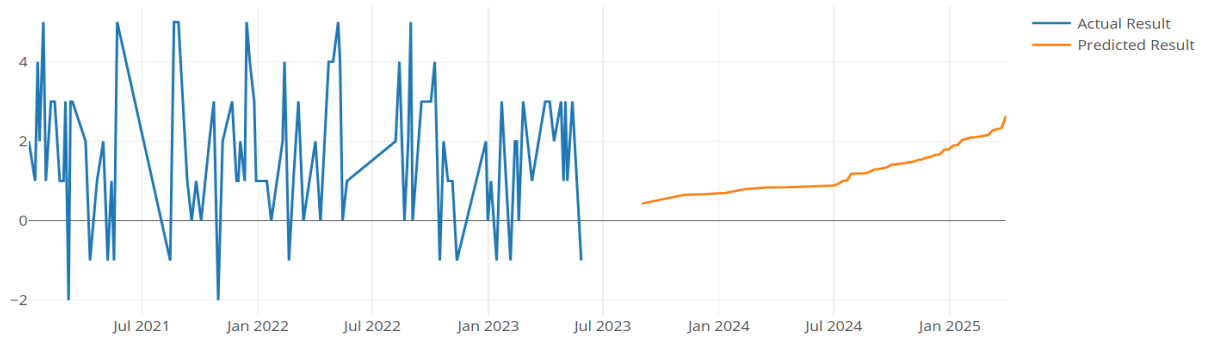
Liverpool's Last 100 games and Next 50 Games Prediction



Manchester Utd's Last 100 games and Next 50 Games Prediction

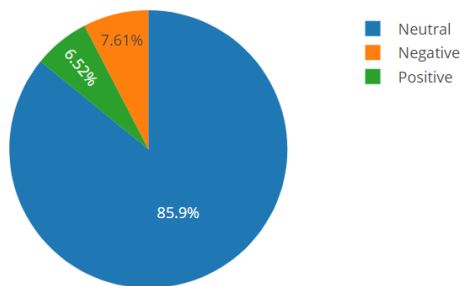


Manchester City's Last 100 games and Next 50 Games Prediction

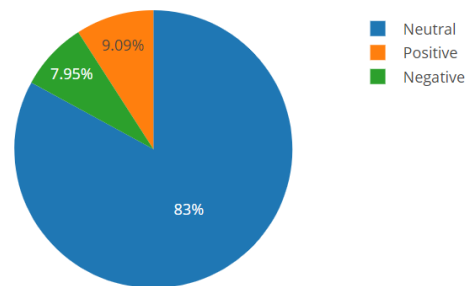


Sentiment Analysis

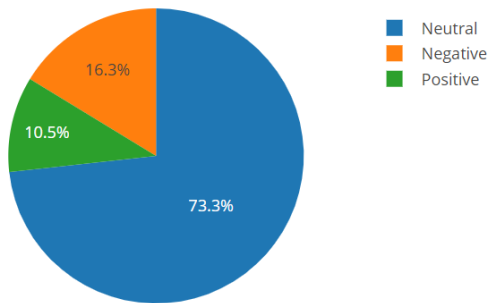
Arsenal Sentiment Analysis



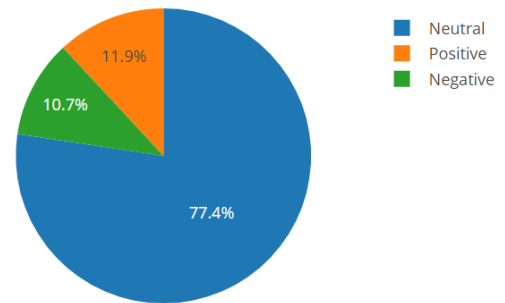
Chelsea Sentiment Analysis



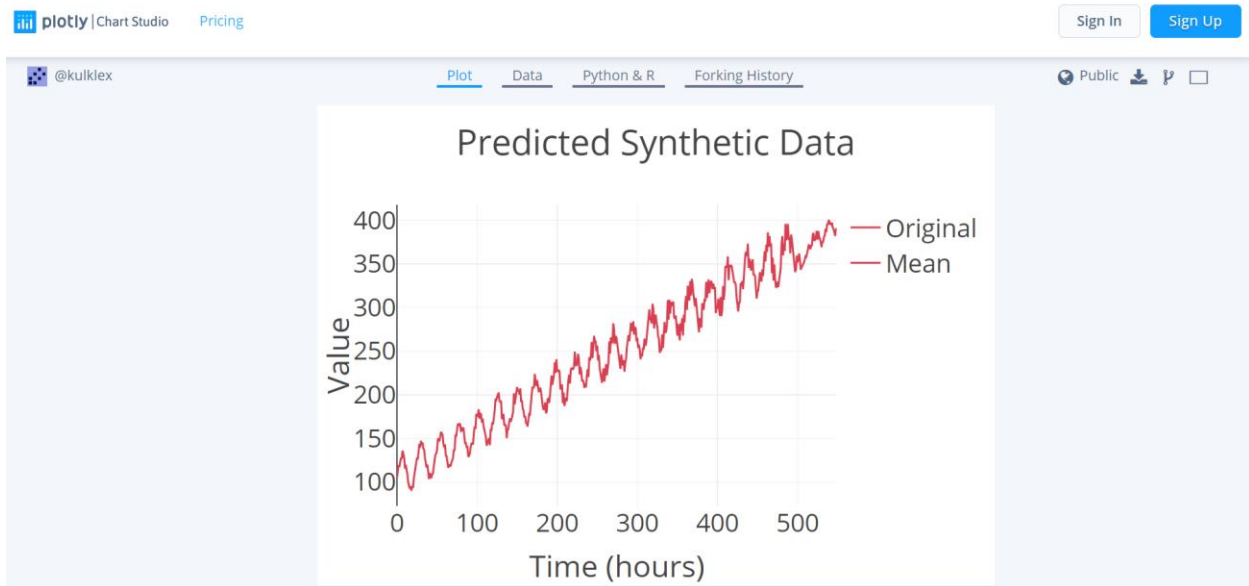
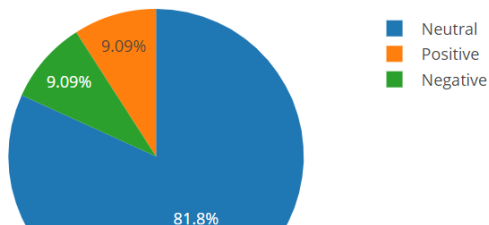
Liverpool Sentiment Analysis



Manchester Utd Sentiment Analysis



Manchester City Sentiment Analysis



@kulklex

Plot Data Python & R Forking History

Public

Synthetic Data for Student M00919866

