

***CST3130 Advanced Web Development
with Big Data Coursework 2 Final
Report***

Jan Ciepiela M00779169



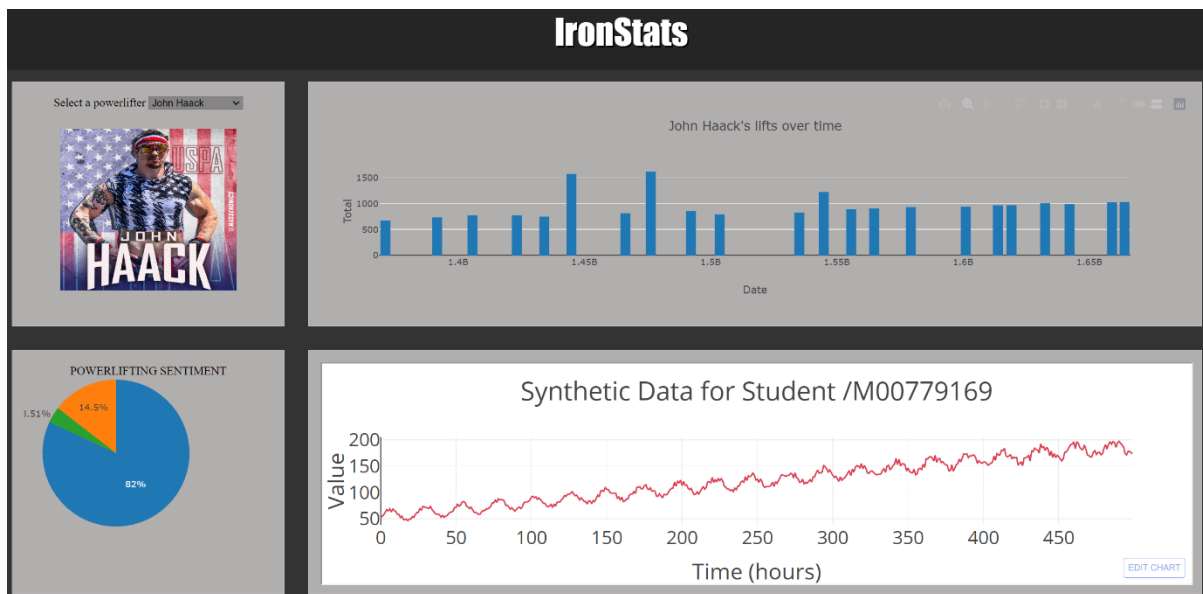
**Middlesex
University
London**

Data visualization website

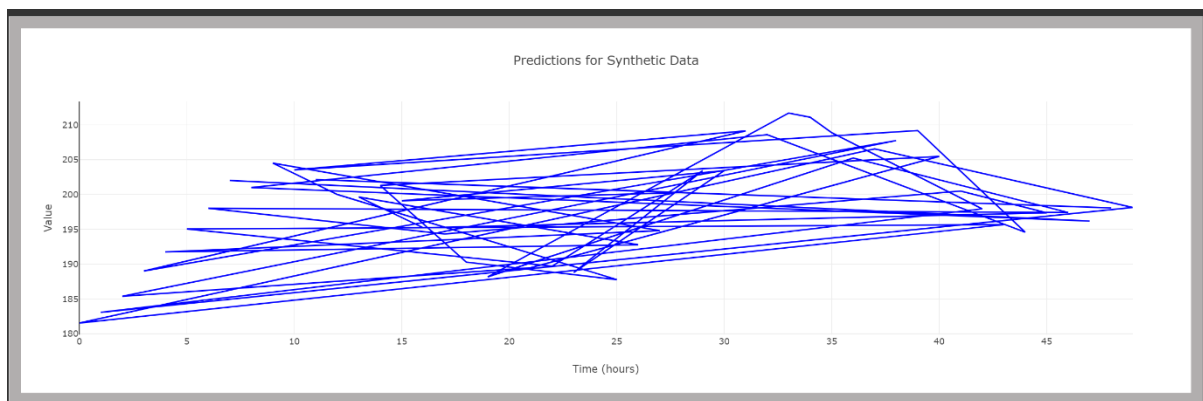
Description of the project

For the purpose of the project, a data visualization website has been produced. The website depicts numerical and sentiment analysis data about powerlifting and 5 chosen athletes, as well as synthetic data and predictions for synthetic data generated using machine learning with AWS SageMaker. Data is visualized using Plotly charts. The flow of the website consists of multiple Lambda functions running in the cloud and sending the data to clients using WebSockets. The frontend is hosted entirely in the cloud using S3 service, as well as backend using serverless technology.

Front-end of the website

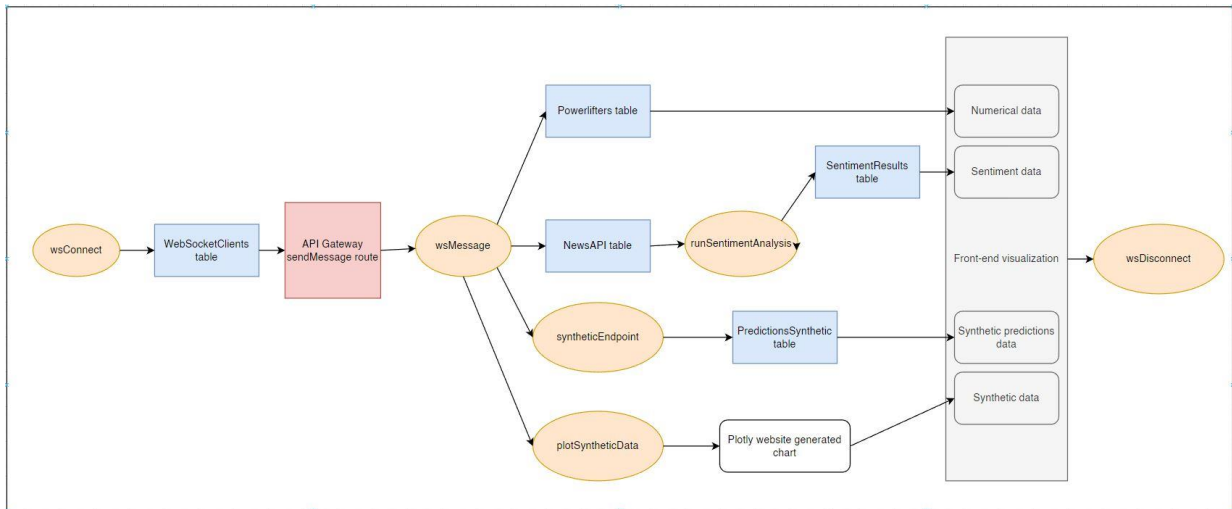


Website consists of a few containers. On the top we can see visualization for the numerical data regarding total weight lifted by certain powerlifters, the selection menu on the left allows us to choose a powerlifter and upon choosing, the new data is loaded from the database. Below the selection menu, the sentiment analysis results is being displayed using pie chart. Right next to it, the synthetic data about the student is shown.



Below in the largest container, the predictions about synthetic data are being displayed.

Architecture of the project



To better describe the flow of data and how lambda functions and database work together to display the data, an architecture diagram has been designed. Upon connection, the id of the client is being sent into the `WebSocketClients` table in a DynamoDB database:

<input type="checkbox"/>	ConnectionId
<input type="checkbox"/>	BiHMrft9IAMCLQA=
<input type="checkbox"/>	Bh54Sd-HIAMCEpg=
<input type="checkbox"/>	BiHQqeSuoAMCLqw=
<input type="checkbox"/>	Bh3gZdOCaAMCLpA=
<input type="checkbox"/>	BiBr6cGvoAMCJzg=
<input type="checkbox"/>	Bh50vf-7oAMCL6w=
<input type="checkbox"/>	Bhs31ecZoAMCJgQ=

When the route “sendMessage” in the API Gateway is called, it triggers the “wsMessage” lambda function, which manages pulling out the data from the Powerlifters table to display the numerical data, the table consists of powerlifters and their best lifts in pounds:

<input type="checkbox"/>	id	name	date	total
<input type="checkbox"/>	187	Zahir Khudayarov	1273795200	937.5
<input type="checkbox"/>	154	John Haack	1535155200	825
<input type="checkbox"/>	7	Blake Lehew	1601078400	815
<input type="checkbox"/>	115	Yury Belkin	1449100800	370
<input type="checkbox"/>	117	Yury Belkin	1438905600	900
<input type="checkbox"/>	47	Blake Lehew	1328832000	562.45
<input type="checkbox"/>	184	Zahir Khudayarov	1314316800	250

When it comes to displaying sentiment analysis results, when the new data is pushed into NewsAPI table, which contains articles about powerlifting,

<input type="checkbox"/>	title	description	publishedAt	source	url
<input type="checkbox"/>	Entrepreneurs Aren't ...	There's a wide...	2023-02-06T...	Forbes	https://www.forbes.com/sites/jodiecook/...
<input type="checkbox"/>	Powerlifter Hunter He...	Hunter Hende...	2023-02-20T...	FitnessVolt...	https://fitnessvolt.com/hunter-henderson...
<input type="checkbox"/>	John Cena Spotted Ro...	The former W...	2023-02-11T...	Freerepubli...	https://freerepublic.com/focus/f-backroo...
<input type="checkbox"/>	Powerlifter Blake Leh...	Blake Lehew ...	2023-02-28T...	Breakingm...	https://breakingmuscle.com/blake-lehew...

the lambda function called “runSentimentAnalysis” is called and using Comprehend, runs the sentiment analysis on powerlifting based on the contents of that table, and then pushes the results into the SentimentResults table:

<input type="checkbox"/>	title	mixed	negative	neutral	positive	sentiment
<input type="checkbox"/>	Powerlifter Hunter He...	0.0000418...	0.0002116...	0.8082252...	0.1915213...	NEUTRAL
<input type="checkbox"/>	Jesus Olivares Demoli...	0.0000848...	0.0560614...	0.9395008...	0.0043529...	NEUTRAL
<input type="checkbox"/>	How to Start Training ...	0.0001281...	0.0022308...	0.5981884...	0.3994525...	NEUTRAL
<input type="checkbox"/>	Powerlifter Ray Willia...	0.0000132...	0.0005145...	0.8859257...	0.1135464...	NEUTRAL
<input type="checkbox"/>	Lynx coach Cheryl Ree...	0.0001445...	0.0012984...	0.4759381...	0.5226187...	POSITIVE
<input type="checkbox"/>	Hong Kong demands ...	0.0000539...	0.2039333...	0.7892547...	0.0067579...	NEUTRAL

“wsMessage” also takes care of pulling out this data and then displaying it on the website.

As for the synthetic data, the original set of numbers is being plotted on the Plotly website after calling the lambda function “plotSyntheticData” and then displayed using iFrame.

The predictions about synthetic data are computed DeepAR algorithm with tool AWS SageMaker, which builds and trains the model as well as creates an endpoint, which is called in lambda function “syntheticEndpoint”, which then also pushes the predictions into the PredictionsSynthetic table:

<input type="checkbox"/>	x	▼	y
<input type="checkbox"/>	7		202.0208129883
<input type="checkbox"/>	47		196.1900939941
<input type="checkbox"/>	8		201.0072021484
<input type="checkbox"/>	32		208.6035003662
<input type="checkbox"/>	44		194.5837097168
<input type="checkbox"/>	39		209.2039031982

Then “wsMessage” retrieve the x and y values and sends them to the front-end which draws Plotly chart based on those values.