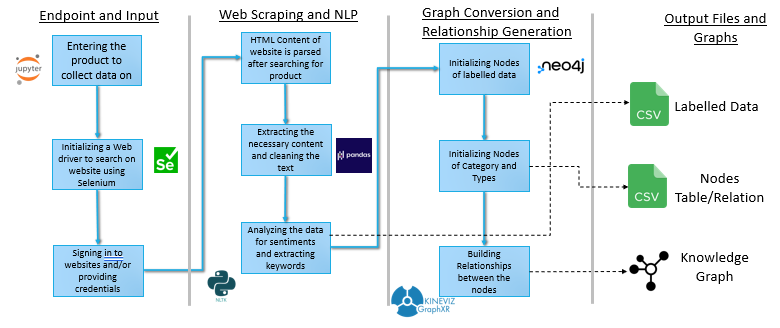
**DATA ANALYSIS USING QUERYABLE KNOWLEDGE GRAPH**

**Introduction:** In today's fiercely competitive tech landscape, success hinges on innovation and understanding customer sentiments. Our research project leverages social media data, sentiment analysis, and knowledge graphs to gain deep insights, identify user communities, and inform strategic decisions. Our main goal is to enhance product market performance.

This initiative positions us as a customer-centric, data-driven leader. We'll explore sentiment analysis, knowledge graphs, and practical applications. Our aim is to thrive in a rapidly evolving market, align with customer expectations, and excel in tech.

**Overview:** To address the challenges, this research project seeks to leverage the power of social media data, sentiment analysis, and knowledge graph technology to create a robust framework for customer-centric decision-making in the technology industry.



**Implementation:**  As first step, we import the libraries into the Jupyter notebook and initialize a web driver using selenium. The web driver is then directed towards the intended page of the website (twitter in this case) and the user is asked to login with his/her credentials. This step is necessary as most social media websites have blocked or have paid incoming API requests.

Once completed, the search bar is passed with the product for which data is needed and, the scraping begins. The driver is automatically scrolled for the tweets that are saved to a list using the sleep function and data is then extracted from each tweet. For this we use the XML path of the HTML tags containing the content of the post. We are then presented with the Username, Handle, likes, retweets, number of comments, and text of the tweet which can be stored to a .csv file.

In the next step, the uncleaned data is converted into cleaned data. The NA values are converted to zero using pandas and the text is converted to lexicon and tokens with the stop words being removed.

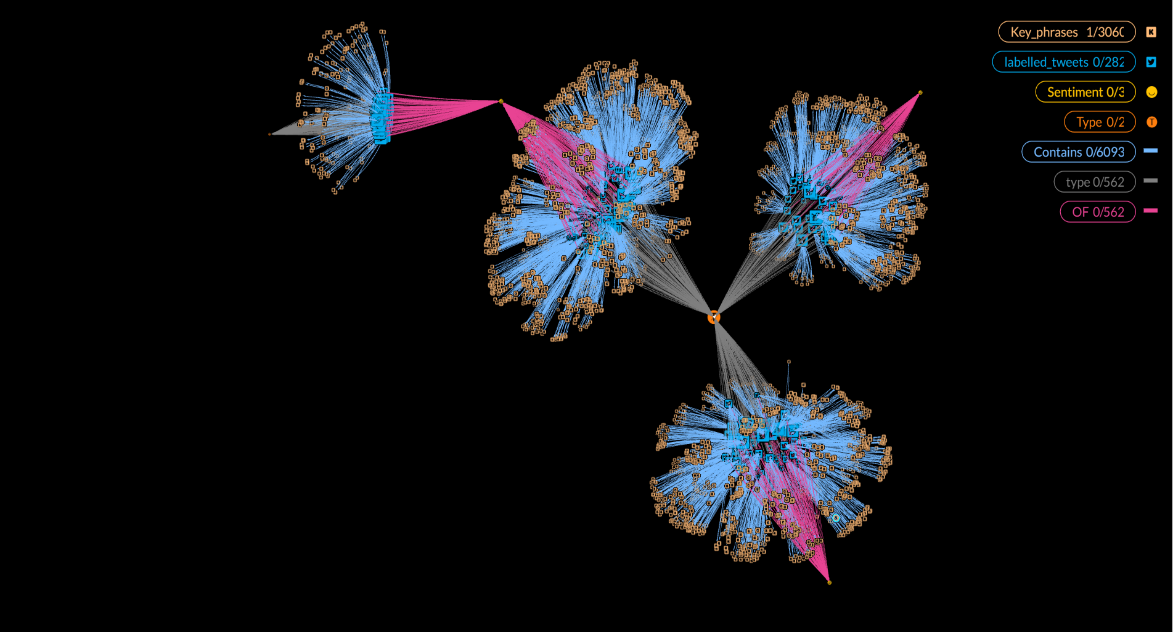
**TextBlob** a library which performs sentiment analysis on a given string by assigning a score to every lexicon, is used. Upon its operation we are presented with a dictionary of scores, i.e., positive, negative, neutral, and compound with values ranging from -1 to 1. This way seems fit as we have unsupervised model to be clustered. For generating which product, the post talks about and what features it mentions, we can use keyword/key phrase generation using **yake** library.

In this each extracted key phrase is assigned a score based on how important it is deemed by the unsupervised model. Thus, as long as keyphrases exist in the text they can be extracted. For this model we shall be extracting the top 20 phrases, but this can be increased as deemed necessary by the developer to generate more information from the post by sacrificing on the fact that some keyphrases may not be useful.

The dataframe with the data is now converted to a knowledge graph. For this we shall be using **neo4j** a free and native graph database. The desktop version is used and is linked to the jupyter notebook using its associated library. The nodes are loaded into the database using cypher queries which are also used to create nodes of types – replies and original, and of the sentiments to which the data has been categorized with their associated relationships.

The built knowledge graph is visualized using **GraphXR** and the nodes are changed to appropriate color and placement in a 3d space. The complete data of nodes and relationship is stored into .csv files also.

**Conclusion:**

****

This graph consists of:

- Data of tweets (Sea blue twitter nodes)

- Type of tweets, i.e., Replies and original (Nodes in orange connected with grey)

- Sentiments of Complaint, Appreciation and Neutral (Connected to nodes in pink)

- Key phrases attached to respective tweet (Connected to tweets in light blue)

**KNOWLEDGE GRAPH OF TWEETS ON HP PRINTERS**