Live Project Enhancement
&
Fluid Dashboarding

A

Project Report

Submitted in the partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Electronics & Computer Engineering

Submitted by

Ravideep Singh

Roll No - 101715128

Faculty Mentor
Dr. Rana Pratap Yadav
Assistant Professor
ECED,TIET,
Patiala

Industry Mentor
Mr. Paresh Pradhan
Consultant
Absolutdata,
Gurugram

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT
TIET, PATIALA-147004, PUNJAB
INDIA
June 2021

Certificate

Certified that project entitled **"Live Project Enhancement & Fluid Dashboarding"** which is being submitted by **Ravideep Singh (University Registration No.** 101715128**)** to the **Department of Electronics and Communication Engineering, TIET, Patiala, Punjab,** is a record of project work carried out by him under guidance and supervision of Mr. Paresh Pardhan. The matter presented in this project report does not incorporate without acknowledgment any material previously published or written by any other person except where due reference is made in the text.

Ravideep Singh
101715128

Faculty Mentor
Dr. Rana Pratap Yadav
Assistant Professor
ECED,TIET,
Patiala

Industry Mentor
Mr. Paresh Pradhan
Consultant
Absolutdata,
Gurugram

Acknowledgement

I would like to express my gratitude to my mentor Mr. Paresh Pradhan. He has been of extraordinary assistance in my endeavor, and a fundamental asset of specialized information. He is genuinely an astounding guide to have.

I am likewise grateful to Mr. Sahil Desai, Mr. Abhimanyu Saraf, and the whole workforce at Absolutdata Research and Analytics, who dedicated their significant time and helped me in every single imaginable way towards the fruitful fulfillment of this venture. I thank each one of the individuals who have contributed either legitimately or by implication towards this venture.

I would also like to thank Dr. Rana Pratap Yadav for timely updating with the internship activities even in the difficult times due to the pandemic.

In conclusion, I also want to thank my family for their immovable love and consolation. They constantly wanted the best for me and I appreciate their assurance and penance.

Abstract

NAVIK SIGNALS drives innovation by providing a comprehensive overview of different competitors, customers, and consumers. The product keeps a track of over 100 types of data from millions of sources and generates business insights, creating a collective view of markets and products. The service is developed using various modules that use Machine Learning and AI technologies, which help in solving essential business problems.

Also, the field of Dashboards and similar products face some challenges in producing important insights. Less hyper-personalization and various other factors are responsible for less effectiveness of the classical dashboards. The introduction of Fluid Dashboards in the market has changed the face of business analytics by providing personalized insights and KPIs. Also, various tasks are being handled by such dashboards, solving the problems faced by organizations.

List of Figures

List of Tables

ACRONYMS AND ABBREVIATIONS

AI	**A**rtificial **I**ntelligence
ML	**M**achine **L**earning
CPG	**C**onsumer **P**ackages **G**oods
KPI	**K**ey **P**erformance **I**ndex
NLP	**N**atural **L**anguage **P**rocessing
RCA	**R**oot **C**ause **A**nalysis
API	**A**pplication **P**rogramming **I**nterface
BERT	**B**idirectional **E**ncoder **R**epresentations from **T**ransformers
NER	**N**ame **E**ntity **R**ecognition
DL	**D**eep **L**earning
SVM	**S**upport **V**ector **M**achine
PoC	**P**roof **o**f **C**oncept
ARIMA	**Auto-Regressive** **I**ntegrated **M**oving **A**verage
SHAP	**SH**apley **A**dditive ex**P**lanations

CONTENT

CHAPTER 1
INTRODUCTION

1.1. NAVIK SIGNALS

1.1.1. Background

NAVIK SIGNALS [1] is a service by Absolutdata that drives innovation by delivering a thorough perspective of competitors, customers, and consumers. It is able to track more than 100 data types from millions of exogenous sources, extract business insights, and create an integrated view of market and category.

The service includes intelligent layers powered by Artificial Intelligence and human expertise to add key insights and signals to an organization's information. Machine learning models are capable of discovering growth prospects and potential problems.

1.1.2. Modules to be enhanced

NAVIK SIGNALS is composed of multiple modules that help the service to cater to the needs of an organization effectively. To improve the service, there is always a need to regularly enhance and develop modules as the field of AI and ML is advancing rapidly.

Some of the modules that were enhanced and developed as the part of enhancement of the service are as follows:

- Annual Report Summarization development
- Entity Standardization and Unknown Entity Extraction enhancement
- Ontology Keywords Variations enhancement
- Twitter Data extraction for Real-Time Analytics development
- Text classification for Intent models enhancement

1.1.3. Services offered

NAVIK SIGNALS is majorly used by the industries like foodservice, e-commerce, healthcare, CPG, personal care, media, travel, hospitality, etc. It provides the ability to track competition, consumers, customers, and innovation. Also, using Natural Language Processing, it is able to answer business-critical questions explaining the market scenario and identify hidden opportunities. It is able to provide a 360° view of the market by providing information to marketing, sales, product, and executive management teams.

1.2. FLUID DASHBOARDING

1.2.1. Background

Classical Dashboards are used to provide insights into KPIs. They are used for information management and business intelligence. They are customizable as per the needs of the individual. But there are some limitations associated with classical dashboards. They are made of components called pagelets [2], which are less interactive. Also, there is no option for hyper-personalization. Like in an organization, people at different levels need to see different insights that serve their purpose. Another problem is that in classical dashboards, KPIs don't change according to time. So to solve this problem, ***Fluid Dashboards*** are a great alternative.

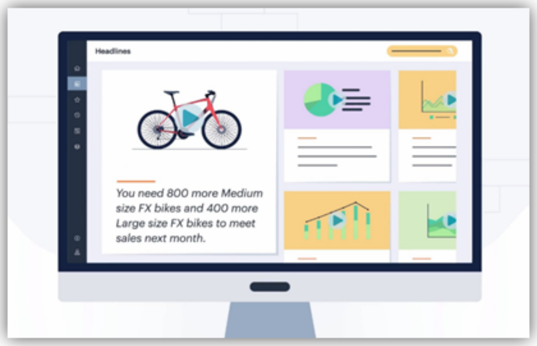

Headlines
You need 800 more Medium size FX bikes and 400 more Large size FX bikes to meet sales next month.

Figure 1.1. - Fluid Dashboard [3]

Fluid Dashboards provide dynamic data stories with a hyper-personalized experience that replaces excessive clicking and exploration, saving time and money. It leverages technologies like augmented analytics, NLP, anomaly detection, etc. The fluid dashboard provides more relevant insights based on the user's context, role, or use.

1.2.2. Modules developed

Fluid Dashboard is basically a combination of various modules that are clubbed together so that users can get maximum benefit from it. As this initiative is in a PoC stage, we have selected the task of Anomaly detection [4] and insights generation for making a demo of Fluid Dashboard that can be shown to the client and necessary changes can be made according to customer's demand. So, modules developed for making the dashboard are as follows:

- Anomaly Detection module development
- Root Cause Analysis (RCA)
- Chart to Text module development
- Role-based insights generation module

CHAPTER 2

THEORY AND FINDINGS

2.1. RESEARCH FOR ENHANCEMENT OF NAVIK SIGNALS MODULES

2.1.1. Theory associated with the module

- **Annual Report Summarization:**

  A Text Summarizer is a system that allows the user to retrieve important information from a large text by shortening the content into a small text. The intention is to create a fluent summary consisting of only the main points of the large text.

  In today's world, where data is considered as the new 'Oil', Text Summarization is becoming increasingly important. Organizations are generating an enormous amount of data in the form of annual reports. This huge amount of data is difficult to interpret as it consumes a lot of valuable time of the organization. Therefore it becomes necessary for the companies to generate valuable insights from these reports without devoting their precious time. Here, text summarization plays a crucial role by reducing the reading time and accelerating the process of information retrieval.

  The project was divided into multiple stages. As a part of this project, research was conducted into the field of NLP and text summarization. Multiple techniques were implemented for the summarization of a large text, also known as a corpus. These techniques were then compared with each other on various factors and the best one was chosen for the task.

- **Entity Standardization and Unknown Entity Extraction:**

  In raw data, many times there are some entities with some variation in the names, although they belong to the same entity. For example, in a sentence, the word 'Apple' can occur in various forms like 'Apple corporation', 'Apple Inc.', etc. So to bring uniformity in the data, there is a need for standardization of these Entity names. Also, there can be some unknown entities that may prove as an important opportunity. So unknown entities cannot be misinterpreted.

- **Ontology Keyword Variations:**

  In a long text, for a particular keyword, there may be various variations present. A single keyword is written all over the text in various forms that makes it a little difficult to interpret. So the module developed here makes a dictionary of keywords with their variations so that we can look them up in a dictionary and identify the standard keyword for a particular variation. Description of the data can not be provided due to confidentiality.

- **Real-Time Analytics - Twitter data extraction:**

  Twitter is one of the foremost and most popular microblogging websites, where one can place their thoughts on the internet. Several organizations use this to advertise their products, run campaigns, and far more. The data available on Twitter can extremely facilitate us to develop bots, automation tools that can help firms to keep track of their competitors in real-time. For this, Twitter provides an Application Programming Interface additionally known as API, to extract information like tweets, likes, shares, retweets, etc., which can be analyzed to describe different aspects of the market.

- **Text classification for Intent models:**

  The data that we have consisted of news headlines that need to be analyzed so that we can identify the competitors, customers, and hidden opportunities. These headlines can be classified under various titles to make analysis more easy and efficient. To do this, Text classification algorithms are applied to the dataset available which assigns a number of predefined categories to the text. These algorithms can be used to organize, and categorize any type of text across the web.

2.1.2. Identification of the techniques:

- **Annual Report Summarization**

This section describes the methodologies used and discusses their advantages and limitations. Significant research has been performed in the field of Text Summarization to improve the results. Some of the approaches that we used during this project are as follows:

❖ **BART**

BART [6] is a denoising autoencoder model used for the pretraining sequence-to-sequence model developed by Facebook. It can perform multiple tasks and performs well for abstractive summarization problems. It is capable of producing coherent grammatical sentences that capture the sense of the text it should summarize. It uses a standard Transformer-based architecture that generalizes BERT [7](due to the bidirectional encoder), GPT [8] (with the left-to-right decoder), and other recent pre-training schemes.

Limitations of this approach include varied processing times and high memory usage. These limitations make this model less efficient when the size of the input text increases.

❖ **LexRank**

LexRank [9] is an unsupervised graph-based approach for automatic text summarization. It is based on the centrality scoring of the sentences. If one sentence is very similar to many others, it will likely be a sentence of great importance. The importance of the sentence is also based on the importance of the sentences recommending it.

LexRank uses IDF-modified Cosine as the similarity measure between two sentences. This similarity is used as the weight of the graph edge between two sentences. LexRank also incorporates an intelligent post-processing step which makes sure that top sentences chosen for the summary are not too similar to each other.

Thus, to get ranked highly and placed in a summary, a sentence must be similar to many sentences that are in turn also similar to many other sentences. This makes intuitive sense and allows the algorithms to be applied to any arbitrary new text.

Being an extractive summarization model, it is very fast and it maintains redundancy. It also has some limitations. One of the main disadvantages is the Dangling anaphoric problem [10]. The summarized expression produced whose antecedent has not been included in the summary, cannot be interpreted or are interpreted incorrectly.

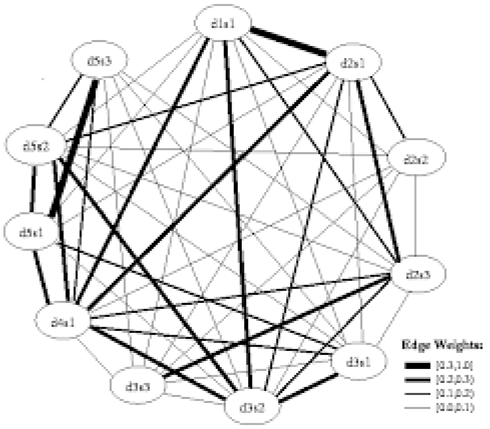


Figure 2.1. Lexrank algorithm

❖ Sentence Scoring

Sentence scoring [11] using term frequency is one of the extractive approaches for text summarization. Here we assign weights to each word based on the frequency of the word in the passage. Using the weights assigned to each word, we will create a score for each sentence. In the end, we will be taking the score of the top `N` sentences for the summary.

Although the method performs very well, there are some issues due to heavy reliance on the vocabulary. A common issue is that the words which are identical in meaning, are not leveraged separately in the word frequency dictionary.

❖ TextRank

TextRank [12] is an algorithm that uses an extractive approach and is an unsupervised graph-based summarization technique. It is based on the PageRank algorithm [13], introduced by Larry Page**,** one of the co-founders of Google.

Firstly, it extracts all the sentences from a text document. A graph is created out of the extracted sentences, where nodes represent sentences and edges represent the similarity between any two nodes. The score of each node is found by iterating the PageRank algorithm until convergence. At last, sentences are sorted in a descending order based upon the scores. The first n sentences are chosen for the summary preparation.

- **Entity Standardization and Unknown Entity Extraction:**

As the module consists of multiple code functions, there was a need to identify the code which had a scope of improvement, increasing the accuracy and efficiency of the model. The first step consisted of selecting a Name Entity Recognition (NER) [14] model.

There are various types of models associated with NER:

- ❖ BERT NER:
  - ➢ BERT stands for Bidirectional Encoder Representation of Transformers
  - ➢ General-purpose language model that can be fine-tuned for different tasks like Text classification, question answering, and many more
  - ➢ State-of-the-Art model for NLP use cases.

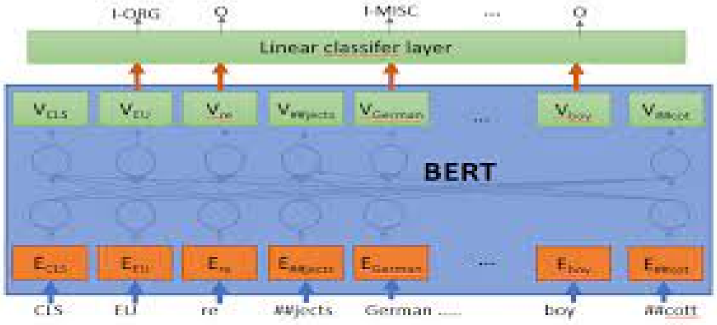


Figure 2.2. BERT NER model architecture

- ❖ spaCy NER:
    - ➢ Fastest NLP framework in Python
    - ➢ Open-source library for Python usage
    - ➢ Features NER, POS tagging, word vectors, etc.
    - ➢ Based on transformers architecture

The standardization module also consists of functions that clean the dataset for better efficiency and usage. Also, the module uses text similarity algorithms to find similarities between various keywords so that we are able to standardize the keywords with accuracy and precision. There are some of the algorithms that can be used for this task:

- ❖ Levenshtein Distance:
    - ➢ Calculates the distance between two unequal sizes of strings
    - ➢ The distance value describes a minimal number of alteration like addition, deletion, substitution, etc. required to make source and target strings equal
    - ➢ Lesser the distance between 2 strings, the more similar the strings
- ❖ Cosine similarity:
    - ➢ Measures similarity between two vectors of inner product space
    - ➢ Measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction.
    - ➢ Used to measure similarity in an analysis of textual data
- ❖ FuzzyWuzzy:
    - ➢ Python library used for string matching
    - ➢ Scores the similarity out of 100, more the value, similar the strings
    - ➢ Consists of powerful functions to help in string matching in complex scenarios.

- **Ontology Keywords variations**

There are multiple techniques that are able to generate variations of a particular keyword in Python. Some of them are described below:

- ❖ Rule-based variation generation:
    - ➢ Based on rules according to the English language
    - ➢ Take care of all the tenses like - Present, Past, Future.
- ❖ Inflect and Word_form library:
    - ➢ Inflect library generates plural, singular nouns, and articles
    - ➢ Word_form library generates all possible forms of an English word
    - ➢ It can conjugate verbs and can join various parts of speeches

- **Real-Time Analytics - Twitter Data extraction**

An important part of Real-Time Analytics is real-time data extraction. Twitter, being an important source of data, can be used to fetch tweets, retweets, likes, etc. This information can be used in various forms like sentiment analysis, opportunistic growth identification, etc.

Methods available for data extraction are:

- ❖ Twint [15]:
    - ➢ Twitter scraping tool that can be used to scrape tweets without the use of API.
    - ➢ Utilizes Twitter's search operators to scrape tweets for particular users, topics, hashtags, trends, etc.
    - ➢ No limitation of rate but faces problem in the legality usage for an organization
- ❖ Tweepy [16]:
    - ➢ Makes use of official Twitter API in the backend
    - ➢ Enables Python to communicate with the Twitter platform and use its API

        - ➢ Open source and easy to use
        - ➢ Takes care of various low-level details about HTTP requests, rate limiting, authentication, serialization, etc.

- **Text classification for Intent models**

  Text classification, being a classic NLP problem, can be solved through various Machine Learning and Deep Learning techniques. As computers are not able to understand human language, each word and sentence is converted into numbers through various techniques as follows:

  - ❖ TF-IDF vectorization [17]:
    - ➢ Based on the frequency method
    - ➢ Takes into account the entire corpus of string while calculating the frequency and score for each word
    - ➢ TF-IDF works by penalizing common words by assigning them lower weights while giving importance to unique words in a particular document
    - ➢ The weights assigned to each word are then combined to form the weight of a sentence which is in the form of numbers that a machine can understand
  - ❖ Word2vec [18]:
    - ➢ Most popular technique to learn word embedding using shallow neural networks
    - ➢ It not only allocates weights according to the frequency of the word but also considers the position of the word in the sentence.
    - ➢ Takes care of dependency of one word on the other
    - ➢ Based on the prediction method

After the conversion of human-made sentences into numerical embeddings, sentences are ready to be trained using ML and DL based algorithms:

- Logistic regression [19]:
  - A statistical model that uses a logistic function to model-dependent variables
  - A supervised learning classification algorithm that predicts the probability of dependent value
  - Only able to predict dichotomous dependent variables

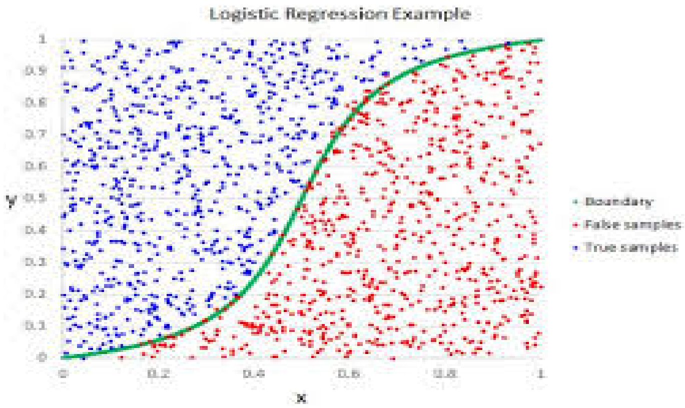


Figure 2.3. Classification using Logistic regression

- Support Vector Machine classifier [20]:
  - Set of supervised learning methods used for classification, regression, and outlier detection
  - This algorithm is effective in high dimensional spaces
  - Memory efficient algorithm that can be used for multiclass classification

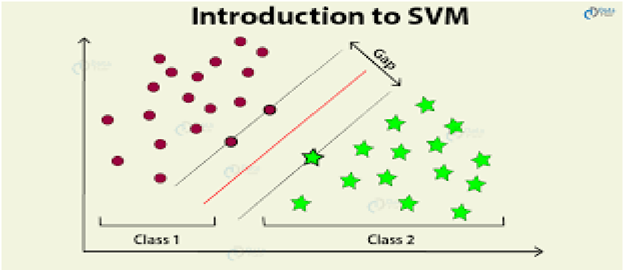


Figure 2.4. Classification using SVM

- XGBoost classification algorithms [21]:
  - Open source python algorithm that provides a regularizing gradient boosting framework
  - Highly flexible, efficient, and portable
  - Implementation of gradient boosted decision trees
  - Can be used for multiclass classification problems

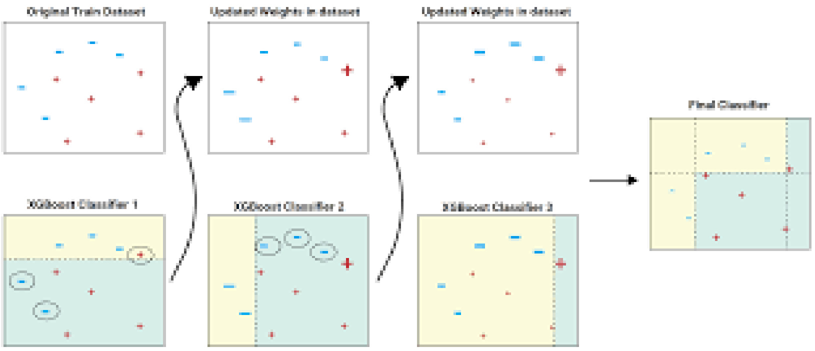


Figure 2.5. Classification using XGBoost

- FastText [22]:
  - Library for learning word embedding and text classification developed by Facebook's AI lab
  - The library allows training the model using a supervised or unsupervised learning approach
  - Multinomial logistic regression is used in the backend and hierarchical softmax increases the computational speed

2.2. RESEARCH RELATED TO FLUID DASHBOARD DEVELOPMENT

2.2.1. Theory associated with the module

Fluid Dashboard is a collection of multiple modules leveraging different technologies to satisfy the needs of the user. Being a PoC, the tasks accomplished consists of modules of Anomaly Detection and insights generation

- **Anomaly Detection:**

  Anomaly detection is an important and complex task in the machine learning field that finds anomalies in a group of temporal data considering various factors like a trend, seasonality, holiday, etc. This problem can be solved through various algorithms available.

- **Root Cause Analysis:**

  To provide credibility to the anomalous points generated by previous algorithms, Root Cause Analysis (RCA) is a perfect way of describing the cause behind any anomaly point.

- **Chart to Text conversion:**

  Sometimes, users are not able to extract insights from a chart due to its complex representation in the case of thousands of data points. The charts are cluttered with the data points. To solve this problem, describing the content of the chart in the form of the text provides more intuitive insights and can easily explain trends, behavior, and many other questions of the users.

- **Insights generation:**

  Fluid Dashboard is all about the depiction of specific insights to the specific user type. As there are different levels in an organization consisting of executives, managers, agents, etc., they all need to see hyper-personalized insights that serve their purpose more effectively. So this module identifies role-specific insights and displays them in the form of interactive charts and other visualization. The KPIs are also depicted using various charts and Natural Language generation.

2.2.1. Identification of the techniques

- **Anomaly Detection:**

  Some of the algorithms that were researched and implemented are as follows:

  - ❖ Isolation forest algorithm [23]:
    - ➢ Unsupervised learning algorithm that works on the principle of isolating the anomalous points in the data using decision trees
    - ➢ It segregates the exceptions by arbitrarily choosing a component from the given arrangement of features and afterward haphazardly choosing a split worth between the most extreme and least values of the selected feature
    - ➢ This algorithm recursively generates partitions on the datasets by randomly selecting a feature and then randomly selecting a split value for the feature.
    - ➢ The issue faced with this algorithm is that it doesn't consider time series components like trend and seasonality.
  - ❖ ARIMA [24]:
    - ➢ ARIMA stands for Auto-Regressive Integrated Moving Average
    - ➢ Based on an approach that several past data points generate a forecast with the added white noise
    - ➢ Anomaly detection is done by building an adjusted model of a signal by using outlier points and checking if it's a better fit than the original model by utilizing t-statistics.
  - ❖ FBProphet [25]:
    - ➢ Open-source library for time series forecasting with great usage in anomaly detection problems
    - ➢ Provides uncertainty intervals along with the prediction.
    - ➢ This interval helps to classify the outlier points with the rest of the data points
    - ➢ Takes care of missing data points and multivariate time series along with trend and seasonality

FBProphet is the best algorithm to use as we have a small dataset (~104 rows) and it is able to perform anomaly detection for multivariate time series and considers time series components like trend and seasonality.

- **Root Cause Analysis:**

  The algorithms that we applied in our PoC is as follows:

  - ❖ XGBoost and SHAP:
    - ➢ XGBoost algorithm is used to train the dataset through a supervised learning method by keeping the anomaly flag as a dependent variable and other variables as an independent variable
    - ➢ After training the model, SHAP [26] - a game-theoretic approach to explaining the output of any machine learning model is used to identify the weightage of all variables to raise the anomaly flag

- **Chart to Text conversion:**

  To convert a chart to its textual explanation we have found 2 methods:

  - ❖ Transformer based model [27]:
    - ➢ Encoder-Decoder based transformer model is trained on a dataset containing chart pictures and gold summaries for sports and financial data
    - ➢ Uses self-attention mechanism to generate latent representations and a binary prediction layer to predict whether that data should be included or not
    - ➢ Positional embedding is added to maintain the order of the information.
    - ➢ Variable substitution is performed so that there is no alteration

in the actual figures

- Rule-based approach:
  - This is a naive approach to convert chart to text
  - This involves using *if-else* rules and hardcoded language
  - Analyzing changes and trend in the values and looking for their derivative to figure out slope helps to generate a human-like text description

- **Insights generation:**

  For an organization, dashboards need to be intuitive and should display KPIs and insights so that companies can identify the problems or the opportunities. Visualizing the data is one of the great ways to make data more communicable and easy to understand. Some of the insights and KPIs that are essential for a company are:

  - Sales trend
  - Total Sales
  - Sales to date
  - Percentage change in sales
  - Most and least affected markets/products
  - Average order value
  - Sales by region
  - Frequency of purchase

CHAPTER 3
PROJECT DEVELOPMENT

3.1. ENHANCEMENT AND DEVELOPMENT OF NAVIK SIGNALS MODULES

3.1.1. Development - Annual Report Summarization

There are three key takeaways from the research conducted in this field :

- Extraction-based summarization techniques are faster than Abstractive summarization techniques.
- Abstractive summarization methods like BART are not able to capture numerical figures well while summarizing large paragraphs
- Despite LexRank and TextRank using the same base algorithm, the key difference is that in TextRank, weights of the edges are taken as unit weight, while in LexRank, weights are calculated using a similarity matrix. Also, LexRank takes the position and length of sentences into account, in contrast to the TextRank algorithm.

As an approach to summarize the annual reports, we would be doing extractive summarization instead of abstractive summarization as it is extremely fast. Also, while comparing TextRank and LexRank summarization, we come to know that both perform well in the summarization task but sometimes, one performs marginally better than the other. This depends upon the type of data we are having. These algorithms perform far better than Sentence Scoring algorithm as it captures the crucial information more effectively. Also, numerical data like costs are well documented in the summary.

A good practice is to first use the Name Entity Recognition (NER) model. This can be used to extract only those sentences that are essential for our results. This model will extract the sentences from the report, which includes words related to money, dates, laws, organizations, etc. These keywords will help us to find the crux of the report. Once all the sentences are extracted, we can run both TextRank and LexRank summarization techniques and choose one, which gives a more satisfactory result.

3.1.2. Enhancement - Entity Standardization and Unknown Entity Extraction

The method adopted for solving this problem involves cleaning all entity names before mapping them to a list of known entities. The entity is replaced with the known entity for successful mapping, and the cleaned variant of the name is used for unsuccessful mapping. Also, it extracts the entities that are not known earlier.

The process involves 4 steps:

Firstly, BERT-NER: a name entity recognition algorithm is used for extracting the required entities from the sentences available. A particular set of entities are selected from the output and the rest are discarded.

Secondly, entity names extracted from the previous step are cleaned. For this process, all the numerical values, special characters, and extra delimiters are removed from the entity's name, as these are not required. Also, generic keywords like corporation, Ltd, Pvt Ltd., etc are removed from the name. Also, there are some junk entity names that are not prevalent according to the use case of this project. So these names, consisting of words like food, aerospace, defense, etc. are removed from the output.

The previous step is then followed by mapping these extracted words with known entities. This mapping is done using string matching algorithms. Using a fuzzy ratio (i.e. token_sort_ratio), this process is accomplished. If the value of the ratio is greater than or equal to 0.85, then the entity is mapped to the known entity.

Lastly, unknown entities are extracted by comparing the retrieved entities against the list of known entities. If there is no match, the entity is tagged as an unknown entity

3.1.3. Enhancement - Ontology Keywords Variations

This module includes functions to generate different variations of keywords. The variations include - present and past tense of different words in the keyword, different forms of nouns and verbs, etc.
The process is as follows:

First, we generate variations of each word in a particular keyword. A keyword is converted into a list of words. Different forms of each word are extracted using libraries called word_forms and inflect. Then, a list of lists is generated with all the variations of the words present in the keywords. Some of the stopwords like and, or, for, etc. are ignored as there are no variations for these words.

Secondly, we generate variations of a keyword. The list of lists for the keyword generated in the previous step is used to find all the combinations possible, of different variations of each word using a recursive function. All the variations are generated in the form of a dictionary named key2val. This dictionary is used to tag multiple variations of keywords to its standard topic.

3.1.4. Development - Twitter Data Extraction for Real-Time Analytics

To access this API using python script, there are few libraries available. But the most famous one is Tweepy. It is a package that helps developers to smoothly access Twitter's endpoints. Hassles like HTTP requests, authentications, serializations, etc. are handled by this package.

To access the data from Twitter, the process is as follows:

- Install Tweepy package which can be done easily by pip install tweepy

- API uses OAuth for authentication purposes as it is more secure and your password remains hidden. For credentials, you need to make a Twitter Developer Account.

- Instantiate the API by using credentials from the previous step. After that, an API object is created

- This object created from Tweepy.API class helps to access information about tweets, likes, retweets, etc. There are different methods that can fetch tweets according to keywords or usernames

- The fetched tweets can then be cleaned by filtering out unwanted information like retweets, replies, mentions, media (if required).

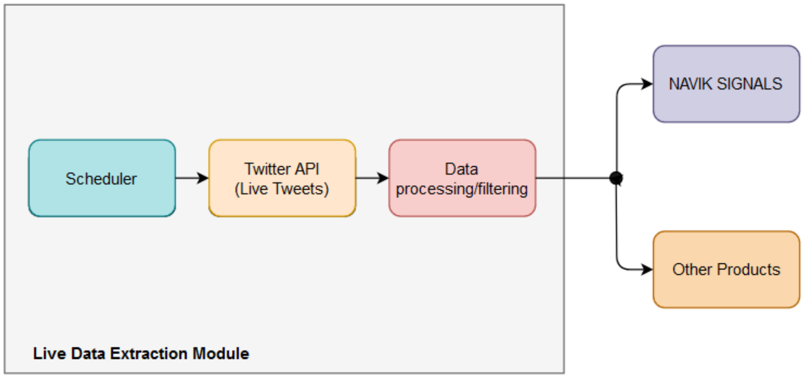


Figure 3.1. - Flowchart for live Twitter data extraction

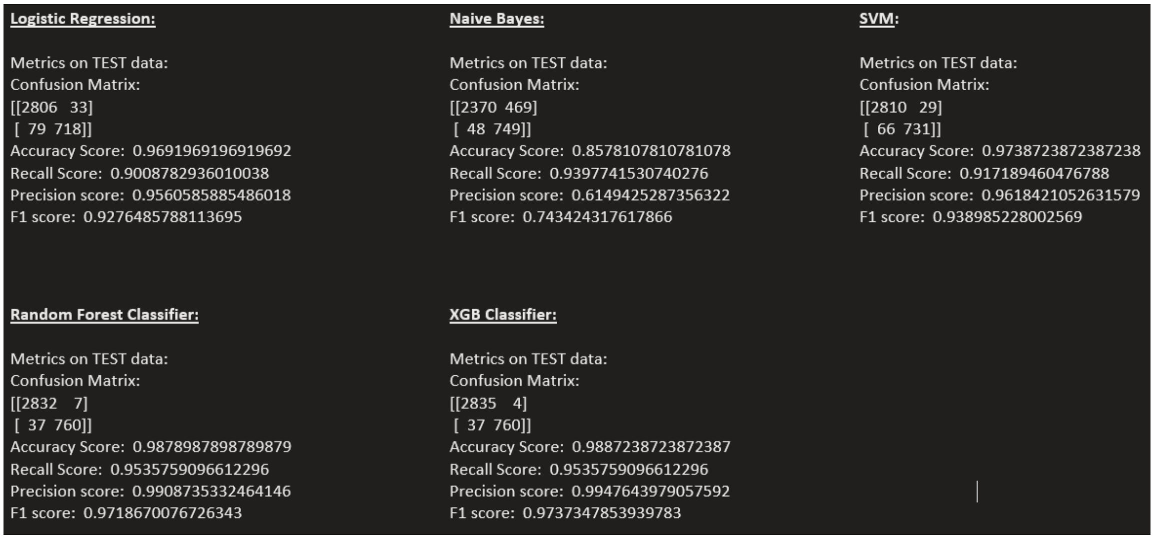
The extracted tweets can then be used to get insights about various fields by applying Natural Language Processing and various other techniques. These insights can then be pushed to databases or email notifications which would increase efficiency.

3.1.5. Enhancement - Text classification for entity models

This module is responsible for classifying different sentences like article titles, news headings, etc. according to a predefined set of categories. To convert sentences into numerical embeddings for easy machine understanding, the following process is followed:

Firstly, each word in the sentence is converted into a numerical embedding using word2vec. All the words are converted into a vector format which is a numeric representation. This numeric representation can then be used for different machine learning algorithms.

Various Machine Learning algorithms are then implemented and then compared on the basis of test metrics as follows:



```
Logistic Regression:

Metrics on TEST data:
Confusion Matrix:
[[2806   33]
 [  79  718]]
Accuracy Score:  0.9691969196919692
Recall Score:  0.9008782936010038
Precision score:  0.9560585885486018
F1 score:  0.9276485788113695
```

```
Naive Bayes:

Metrics on TEST data:
Confusion Matrix:
[[2370  469]
 [  48  749]]
Accuracy Score:  0.8578107810781078
Recall Score:  0.9397741530740276
Precision score:  0.6149425287356322
F1 score:  0.743424317617866
```

```
SVM:

Metrics on TEST data:
Confusion Matrix:
[[2810   29]
 [  66  731]]
Accuracy Score:  0.9738723872387238
Recall Score:  0.917189460476788
Precision score:  0.9618421052631579
F1 score:  0.938985228002569
```

```
Random Forest Classifier:

Metrics on TEST data:
Confusion Matrix:
[[2832    7]
 [  37  760]]
Accuracy Score:  0.9878987898789879
Recall Score:  0.9535759096612296
Precision score:  0.9908735332464146
F1 score:  0.9718670076726343
```

```
XGB Classifier:

Metrics on TEST data:
Confusion Matrix:
[[2835    4]
 [  37  760]]
Accuracy Score:  0.9887238723872387
Recall Score:  0.9535759096612296
Precision score:  0.9947643979057592
F1 score:  0.9737347853939783
```

Figure 3.2. Test metrics of various algorithms

On comparing, we figure out that XGBoost Classifier performs best among all the algorithms. With bigger metric scores. The output of this algorithm is then compared with the present output data which signifies that the results from this algorithm are somewhat similar to the previously used FastText algorithm. As there was no scope for improvement, the old fast text model was not replaced with the XGBoost algorithm.

3.2. DEVELOPMENT OF ANOMALY DETECTION AND RCA FOR FLUID DASHBOARDING

3.2.1. Development - Anomaly Detection module

Detecting Anomalies in a multivariate time series dataset involves following steps to be completed.

Firstly, available sales dataset consists of more than 40 lakh rows with repetition of week dates. These repeated rows are due to multiple sales entries on a single date. To solve this issue, we first convert product codes into their categories and do one-hot encoding on market names, regions, and categories. After this step, we group the dataset according to the date column. As the dataset available is on a weekly basis and for two years, we get around 104 rows on grouping. This dataset is sorted according to the dates in ascending order.

After preparing the data, we perform anomaly detection on the dataset using FBProphet. The parameters for FBProphet include seasonality, interval width, etc. After selecting appropriate parameters, we get the output with a column holding the anomaly flag, 1 for anomaly and 0 for no anomaly.

3.2.2. Development - Root Cause Analysis

After getting anomaly flags in the output of the anomaly detection module, we do root cause analysis to support the anomalous behavior. This is achieved by using XGBoost and SHAP algorithms.

Firstly, we train a model through a supervised learning approach. For this, we prepare the data in the format where the anomaly flag is considered a dependent variable and the rest of the columns are considered as independent variables. After data preparation, we use the XGBoost model on the dataset prepared. This is followed by a model explanation model known as SHAP. This model is primarily used to explain the output of any machine learning model.

Using this, we identify the weight of each independent variable in accordance with the dependent variable. If the weight of a particular variable is higher, we consider that particular variable to be a greater cause for that anomaly flag. Therefore, we select the variable with the highest SHAP value and declare it as the root cause behind that particular anomaly.

3.2.3. Development - Chart to Text using a rule-based approach

As there are some cases where understanding the insights from a cluttered chart can be very difficult. Therefore, we use a chart to text conversion. As there are many methods involved in doing this, we would approach this problem through a rule-based script.

Firstly, we run a model to find the line that fits the best to the data points. This best fit line helps us to identify the trend of the data points. After getting this line, we find a derivative at every point in the trend line. Using a combination of both these, we can generate a hard-coded human-like text.

This is done using if-else rules. For example, if the trend line is going up but its slope/derivative is going down, we can conclude that the overall trend is saturating. Similarly, increasing trend and its derivative signify an exponential growth in the trend. Using all these combinations we generate natural language insights.

Also, converting the root cause and correlation at anomalous points into a natural language is possible using hardcoded strings.

3.2.4. Development - Role-based insights generation

In Fluid Dashboarding, providing hyper-personalized insights is a key feature. For this, it is important to recognize the KPIs and insights according to a specific user and role.

This is achieved by running scripts for each user. For executives, we need to identify overall and important insights. This can be done using data aggregation on the overall dataset. This can result in important insights like Total Sales, Sales to Date, etc.

On a managerial level, insights like purchase frequency, high and low hitting markets, and products are essential. So using simple mathematical formulas and running anomaly detection modules for each category we can easily identify these insights.

At last, agents always need to drill down on the insights provided to upper management like products responsible for anomalous behavior. Therefore, running anomaly detection modules can help to identify the products behaving most anomalously.

CHAPTER 4
RESULTS AND OUTCOMES

4.1. RESULTS ASSOCIATED WITH NAVIK SIGNALS MODULES

4.1.1. Results for Annual Report Summarization

Annual reports are documents with a large number of sentences (~2000 sentences) that describe the financial information of an organization. Summarizing it reduces the number of sentences but still not less enough to mention them in the report due to restriction of word count. You can see a sample of summarization below:

Original Paragraph:-

An Ethereum Foundation researcher, Carl Beekhuizen, in a recent blogpost has said Ethereum is working on a major shift that will help save up to 99.5 per cent of the energy it currently consumes. Notably, Ethereum already uses much lower energy than the most popular cryptocurrency Bitcoin. Ethereum will soon be completing the transition to Proof-of-Stake (PoS) from the Proof-of-Work (PoW) system, according to Beekhuizen. "Digiconomist estimates that Ethereum miners currently consume 44.49 TWh per year which works out to 5.13 gigawatt continuingly. This means that PoS is ~2000x more energy efficient based on the conservative estimates, which reflects a reduction of at least 99.95% in total energy use," he added. The PoS method, already in use by other smaller cryptocurrencies, permits access to new coins based on how many coins a miner already owns; if a miner owns 3% of all coins, they can access only 3% of new coins. This system eliminates the need for energy-intensive number-crunching, because a miner's rate of coin access is a product of their "stake," not of their "work." As a result, proof-of-stake mining software can essentially work on one normal computer, rather than a warehouse of servers, and there is no longer any strategic need to consume an increasing amount of energy. Both Bitcoin and Etheruem currently use the Proof-of-Work model. Beyond the energy consumption issue, in the case of PoW, there are also fears that mining pools, a joint group of cryptocurrency miners, could dominate the mining game in future, leading to security risks and centralisation of mining power.

Summary:-

An Ethereum Foundation researcher, Carl Beekhuizen, in a recent blogpost has said Ethereum is working on a major shift which will help save up to 99.5 per cent of the energy it currently consumes. Digiconomist estimates that Ethereum miners currently consume 44.49 TWh per year which works out to 5.13 gigawatt continuingly. Ethereum will soon be completing the transition to Proof-of-Stake (PoS) from the Proof-of-Work system, according to Beekhuizen. The PoS method, already in use by other smaller cryptocurrencies, permits access to new coins based on how many coins a miner already owns; if a miner owns 3% of all coins, they can access only 3% of new coins.

4.1.2. Results for Entity Standardization and Unknown Entity Extraction

This module is used to standardize the entities and find unknown entities. We apply this module to nearly 70,000 sentences. We used BERT-NER to extract entities present in each sentence. These names are then clean and a standard name replaces the different variations of the entity's name. Also, we extract the unknown entity. Below you can see a sample result out of 70,000 sentences:

- **Sentence:-**
  NCLT directs power distribution firm not to take any coercive action against Sterling Biotech Ltd. which would cost them $ 10 million in this year

- **Entity known:-**
  NCLT

- **BERT-NER:-**
  [('B-ORG', 'NCLT'), ('B-ORG', 'Sterling Biotech Ltd.'), ('B-MONEY', '$ 10 million'), , ('B-DATE', 'this year')]

- **Raw entity:-**
  NCTL||Sterling Biotech Ltd.

- **Entity cleaned:-**
  NCTL||Sterling Biotech Ltd.

- **Entity standardized:-**
  NCTL||Sterling Biotech

- **Unknown entity:-**
  Sterling Biotech

4.1.3. Results for Ontology Keywords Variations

Suppose we have a keyword. For this keyword, we generate various variations and store them in a dictionary so that whenever we encounter a variation. we can track it down to its standard keyword. For reference of how output look like, you can see some results below:

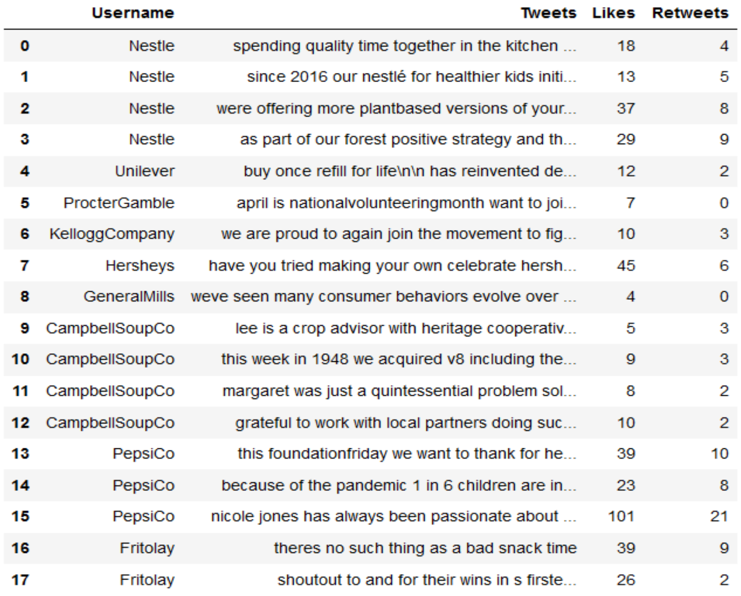
- **Actual Word:-** acute stress
  Variations:- acute stress, acute stresses, acute stressed, acute stressing, acutes stress, acutes stresses, acutes stressed, acutes stressing

- **Actual Word:-** chef-inspired
  Variations:- chef-inspired, chefs inspiring, chefs inspired, chefs inspireds, chefs inspire, chefs inspires, chef inspiring, chef inspired, chef inspireds, chef inspire, chef inspires

- **Actual Word:-** high quality
  Variations:- high quality, high qualities, highs quality, highs qualities

4.1.4. Results for Twitter Data Extraction for Real-Time Analytics

The results shown here are for the usernames/handles searched. Here we are displaying the recent tweets, likes, and retweets for various handles. Similarly, we can also search for multiple keywords as required.

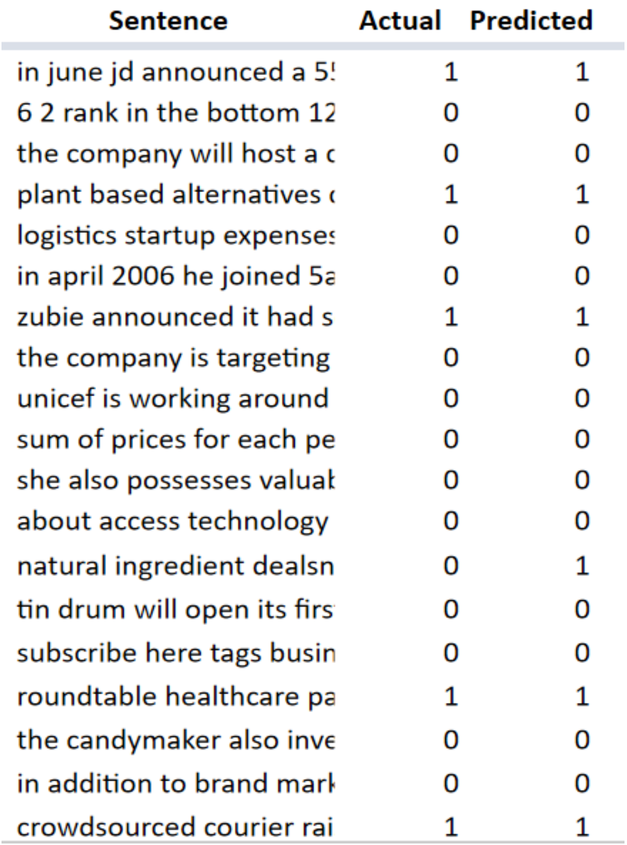


	Username	Tweets	Likes	Retweets
0	Nestle	spending quality time together in the kitchen ...	18	4
1	Nestle	since 2016 our nestlé for healthier kids initi...	13	5
2	Nestle	were offering more plantbased versions of your...	37	8
3	Nestle	as part of our forest positive strategy and th...	29	9
4	Unilever	buy once refill for life\n\n has reinvented de...	12	2
5	ProcterGamble	april is nationalvolunteeringmonth want to joi...	7	0
6	KelloggCompany	we are proud to again join the movement to fig...	10	3
7	Hersheys	have you tried making your own celebrate hersh...	45	6
8	GeneralMills	weve seen many consumer behaviors evolve over ...	4	0
9	CampbellSoupCo	lee is a crop advisor with heritage cooperativ...	5	3
10	CampbellSoupCo	this week in 1948 we acquired v8 including the...	9	3
11	CampbellSoupCo	margaret was just a quintessential problem sol...	8	2
12	CampbellSoupCo	grateful to work with local partners doing suc...	10	2
13	PepsiCo	this foundationfriday we want to thank for he...	39	10
14	PepsiCo	because of the pandemic 1 in 6 children are in...	23	8
15	PepsiCo	nicole jones has always been passionate about ...	101	21
16	Fritolay	theres no such thing as a bad snack time	39	9
17	Fritolay	shoutout to and for their wins in s firste...	26	2

Table 4.1. Live Twitter Data extracted in a dataframe

4.1.5. Results for Test classification for Intent models

The results shown here are the flags that mention whether a particular sentence falls under a particular intent or not. There are basically 5 intents for which ML algorithms are applied to produce results. Some of the results are as follows:



Sentence	Actual	Predicted
in june jd announced a 5	1	1
6 2 rank in the bottom 12	0	0
the company will host a c	0	0
plant based alternatives c	1	1
logistics startup expenses	0	0
in april 2006 he joined 5a	0	0
zubie announced it had s	1	1
the company is targeting	0	0
unicef is working around	0	0
sum of prices for each pe	0	0
she also possesses valuab	0	0
about access technology	0	0
natural ingredient dealsn	0	1
tin drum will open its firs	0	0
subscribe here tags busin	0	0
roundtable healthcare pa	1	1
the candymaker also inve	0	0
in addition to brand mark	0	0
crowdsourced courier rai	1	1

Table 4.2. Result dataframe for classification model

4.2. RESULTS ASSOCIATED WITH FLUID DASHBOARDS

4.2.1. Results for Anomaly Detection module

This module, leveraging FBProphet's capabilities, is able to detect anomalous behavior in a multivariate time series while considering trend and seasonality. Some of the outputs are as shown below:

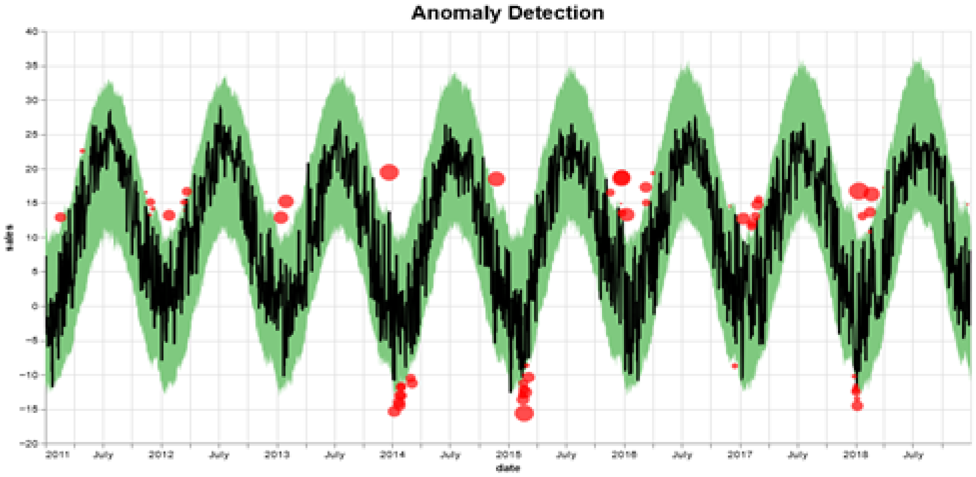


Figure 4.1. Anomaly detection in a sinusoidal wave with noise

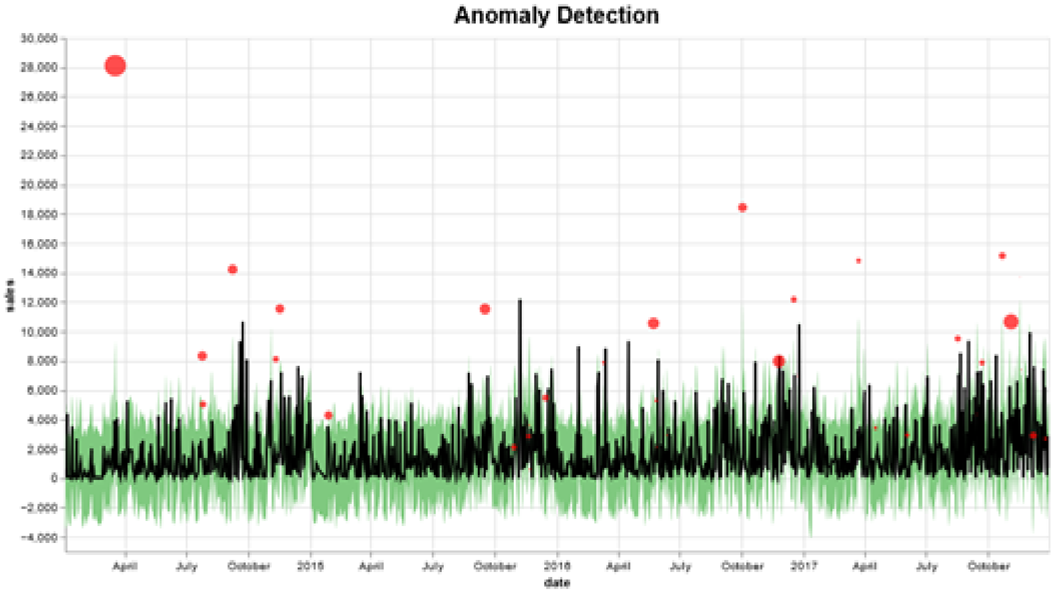
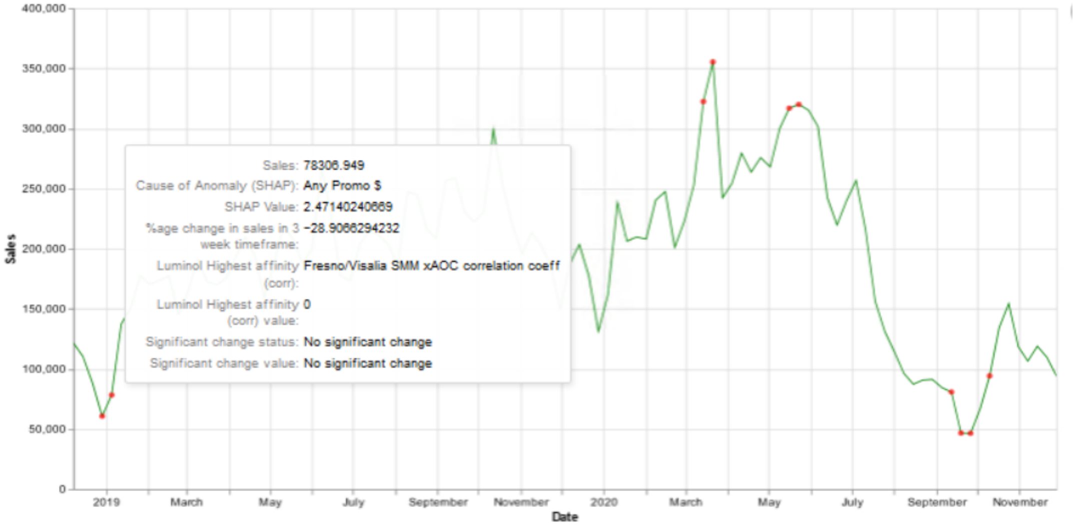


Figure 4.2. Anomaly detection in a sample sales dataset

4.2.2. Results for Root Cause Analysis

This module finds the root cause behind an anomaly using the XGBoost machine learning model that is further explained by SHAP. Furthermore, the correlation of a variable is also calculated with respect to other variables, providing more information about the anomaly point.

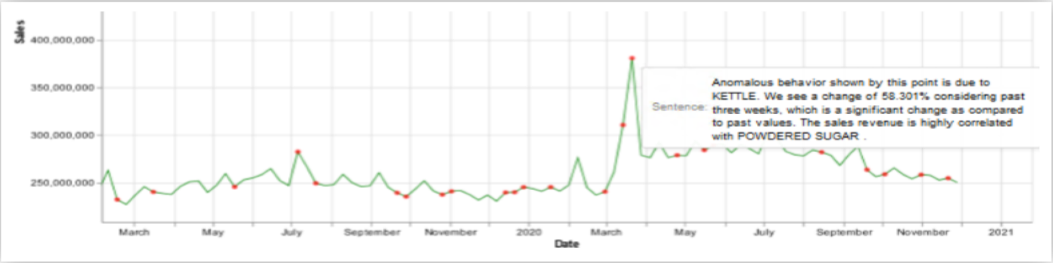


Sales: 78306.949
Cause of Anomaly (SHAP): Any Promo $
SHAP Value: 2.47140240669
%age change in sales in 3 week timeframe: −28.9066294232
Luminol Highest affinity (corr): Fresno/Visalia SMM xAOC correlation coeff
Luminol Highest affinity (corr) value: 0
Significant change status: No significant change
Significant change value: No significant change

Figure 4.3. Root cause analysis on detected anomaly points

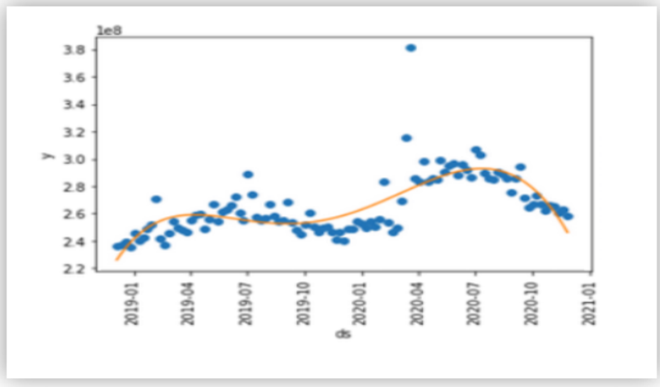
4.2.3. Results for Chart to Text using a rule-based approach

This module is used to convert any chart into a textual format that is easily understandable by humans. The outputs are as follows:



Sentence: Anomalous behavior shown by this point is due to KETTLE. We see a change of 58.301% considering past three weeks, which is a significant change as compared to past values. The sales revenue is highly correlated with POWDERED SUGAR .

Figure 4.4. Root cause explanation in the form of natural language



Initially, there is a rise in the value till it gradually saturates, thereafter sales are accelerating at an increasing growth rate. This is followed by a gradual increase in the values. At last, there is a sharp drop in the values.

Figure 4.5. Trend explanation in the form of natural language

4.2.4. Results for role-based insights

This module is used to generate insights and KPIs in the form of charts and other visualization techniques. The charts vary on the basis of the role of a person. Some of the snapshots for the same are as follows:

<u>For Executive level:</u>

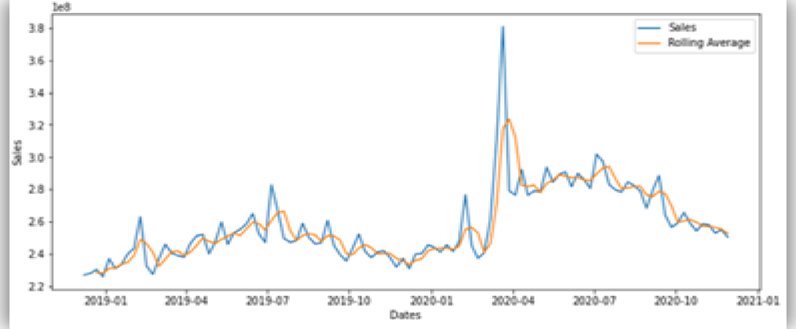
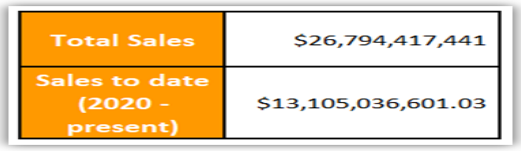


Figure 4.6. Rolling average insight



Total Sales	$26,794,417,441
Sales to date (2020 - present)	$13,105,036,601.03

Figure 4.7. Sample KPIs for sales dataset

For Managerial level:

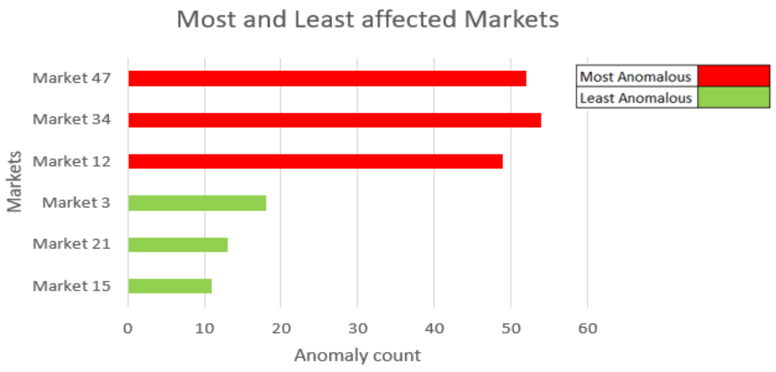


Figure 4.8. Most and Least affected markets plot

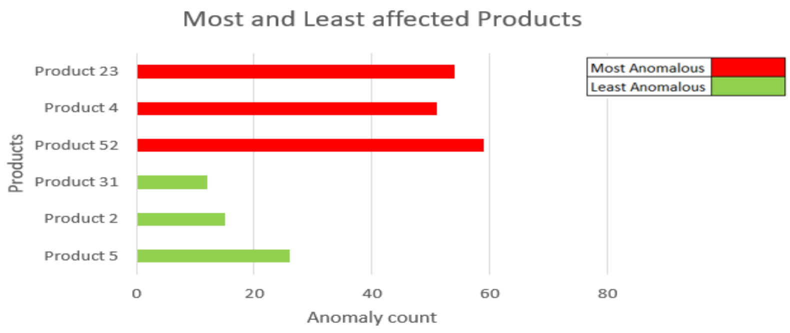


Figure 4.9. Most and Least affected products plot

For Agent level:

KPI	Most and Least affected Products/Markets	UPCs
Most affected product	Product 23 Product 4 Product 52	[1004,1007,1043] [1025,1017,1013] [1001,1061,1078]
Least affected product	Product 31 Product 2 Product 5	[1059,1000,1039] [1071,1087,1021] [1033,1011,1066]
Most affected market	Market 47 Market 34 Market 12	[1055,1037,1015] [1020,1096,1092] [1050,1022,1094]
Least affected market	Market 3 Market 21 Market 15	[1010,1028,1075] [1001,1017,1042] [1061,1026,1000]

Table 4.3. Drilled down insights for agent level

CHAPTER 5
CONCLUSION

Internship at **Absolutdata Research and Analytics** was my first industry experience which guided me about the working of the industry. It really helped me to understand how an organization and business runs. Also, the internship provided me with the opportunity to work on the state-of-the-art algorithms that are prevalent in the market. Some of the work conducted in the field of Natural Language Processing helped me to brush up on my technical skills.

The journey encouraged me to learn more things happening in the data science field. Also, I got the opportunity to experience the various stages during the development of a project. Regular learning helped me to identify the issues in the current code and helped me to solve those problems, thus resulting in better efficiency of the code.

The face-paced environment helped me with my time management skills and also provided me with various opportunities to lead initiatives. At last, this internship helped me to become a technically sound and active person which would further enhance my skills. I hope that in the future, I will be able to make a noticeable impact in the industry.

References:

[1] A. (2020, May 21). *NAVIK SIGNALS-360 degree view of market and category*. Absolutdata. https://www.absolutdata.com/navik-signals-360-degree-view-of-market/

[2] *pagelet*. (2008). TheFreeDictionary.Com. https://encyclopedia2.thefreedictionary.com/pagelet

[3] K. (2020b, July 27). *PeopleSoft- Fluid Dashboards.* Kovaion. https://www.kovaion.com/blog/peoplesoft-fluid-dashboards/

[4] N. Takeishi and T. Yairi, "Anomaly detection from multivariate time-series with sparse representation," 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2014, pp. 2651-2656, doi: 10.1109/SMC.2014.6974327.

[5] Greene, J. (2021, May 14). *The 37 Sales KPIs Every Sales Leader Should Be Measuring | Databox Blog*. Databox. https://databox.com/sales-kpis#:%7E:text=Sales%20KPIs%20are%20measures%20used,goals%2C%20priorities%2C%20and%20objectives.

[6] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V. and Zettlemoyer, L., 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.

[7] Devlin, Jacob, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. "Bert: Pre-training of deep bidirectional transformers for language understanding." *arXiv preprint arXiv:1810.04805* (2018).

[8] Budzianowski, Paweł, and Ivan Vulić. "Hello, it's GPT-2--how can I help you? towards the use of pretrained language models for task-oriented dialogue systems." *arXiv:1907.05774* (2019).

[9] Erkan, Günes, and Dragomir R. Radev. "Lexrank: Graph-based lexical centrality as salience in text summarization." *Journal of artificial intelligence research* 22 (2004): 457-479.

[10] Gupta, Som, and S. K. Gupta. "Abstractive summarization: An overview of the state of the art." *Expert Systems with Applications* 121 (2019): 49-65.

[11] Ferreira, Rafael, Luciano de Souza Cabral, Rafael Dueire Lins, Gabriel Pereira e Silva, Fred Freitas, George DC Cavalcanti, Rinaldo Lima, Steven J. Simske, and Luciano Favaro. "Assessing sentence scoring techniques for extractive text summarization." *Expert systems with applications* 40, no. 14 (2013): 5755-5764.

[12] Mihalcea, Rada, and Paul Tarau. "Textrank: Bringing order into text." In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pp. 404-411. 2004.

[13] Bianchini, Monica, Marco Gori, and Franco Scarselli. "Inside pagerank." *ACM Transactions on Internet Technology (TOIT)* 5, no. 1 (2005): 92-128.

[14] Mansouri, Alireza, Lilly Suriani Affendey, and Ali Mamat. "Named entity recognition approaches." *International Journal of Computer Science and Network Security* 8, no. 2 (2008) pp. 339-344.

[15] *View twint on Snyk Open Source Advisor*. (n.d.). Snyk Advisor. Retrieved June 8, 2021, from https://snyk.io/advisor/python/twint

[16] Roesslein, Joshua. "tweepy Documentation." *Online] http://tweepy. readthedocs. io/en/v3* 5 (2009).

[17] Shi, Congying, Chaojun Xu, and Xiaojiang Yang. "Study of TFIDF algorithm." *Journal of Computer Applications* 29, no. 6 (2009): 167-170.

[18] Church, Kenneth Ward. "Word2Vec." *Natural Language Engineering* 23, no. 1 (2017): 155 - 162.

[19] Dreiseitl, Stephan, and Lucila Ohno-Machado. "Logistic regression and artificial neural network classification models: a methodology review." *Journal of biomedical informatics* 35, no. 5-6 (2002): 352-359.

[20] Noble, William S. "What is a support vector machine?." *Nature biotechnology* 24, no. 12 (2006): 1565-1567.

[21] Chen, Tianqi, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, and Hyunsu Cho. "Xgboost: extreme gradient boosting." *R package version 0.4-2* 1, no. 4 (2015).

[22] Joulin, Armand, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. "Bag of tricks for efficient text classification." *arXiv preprint arXiv:1607.01759* (2016).

[23] Liu, Fei Tony, Kai Ming Ting, and Zhi-Hua Zhou. "Isolation forest." In *2008 eighth ieee international conference on data mining*, pp. 413-422. IEEE, 2008.

[24] Yaacob, Asrul H., Ian KT Tan, Su Fong Chien, and Hon Khi Tan. "Arima based network anomaly detection." In *2010 Second International Conference on Communication Software and Networks*, pp. 205-209. IEEE, 2010.

[25] Kapoor, Krishnam. "A Novel Algorithm for Optimized Real Time Anomaly Detection in Timeseries." *arXiv preprint arXiv:2006.04071* (2020).

[26] Parsa, Amir Bahador, Ali Movahedi, Homa Taghipour, Sybil Derrible, and Abolfazl Kouros Mohammadian. "Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis." *Accident Analysis & Prevention* 136 (2020): 105405.

[27] Obeid, Jason, and Enamul Hoque. "Chart-to-Text: Generating Natural Language Descriptions for Charts by Adapting the Transformer Model." *arXiv preprint arXiv:2010.09142* (2020).