NOVEMBER 2022 M.Sc. in Industrial Engineering MOHAMAD NACİ

REPUBLIC OF TURKEY

GAZİANTEP UNIVERSITY

GRADUATE SCHOOL OF NATURAL & APPLIED SCIENCES

EXPLORING CONCEPT DRIFT IN TECHNOLOGY BY TWEETS MINING

M.Sc. THESIS

IN

INDUSTRIAL ENGINEERING

BY

MOHAMAD NACİ

NOVEMBER 2022

EXPLORING CONCEPT DRIFT IN TECHNOLOGY BY TWEETS MINING

M.Sc. Thesis

in

Industrial Engineering

Gaziantep University

Supervisor

Assoc. Prof. Dr. Alptekin DURMUŞOĞLU

by

Mohamad NACİ

November 2022

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Mohamad NACİ

ABSTRACT

EXPLORING CONCEPT DRIFT IN TECHNOLOGY BY TWEETS MINING

NACİ, Mohamad
M.Sc. in Industrial Engineering
Supervisor: Assoc. Prof. Dr. Alptekin DURMUŞOĞLU
November 2022
59 pages

Over the last decade, a dramatic transform happened in information sources and their use in the digital era. Social media networks have brought a new way of expressing the sentiments of individuals. The matter went beyond being an expression of separate opinions of some individuals, as companies, official institutions and various organizations have pages on the communication sites through which they share various developments, products, opinions, and sometimes even official decisions. Social media has evolved into a medium with a vast quantity of information, allowing users to access the opinions of other users, which can be classified into several sentiment categories, and are gradually taking a crucial role in decision making. Twitter is a microblogging service built to describe what is happening anywhere worldwide, at any moment. It's a fascinating forum for more than 500M messages per day from about 1.3 billion people. Twitter data is short, specific, and easily accessible, that's why it has become one of the best sources for sentimental analysis and knowledge discovery by data streams mining. The fact that the underlying distribution of data may vary over time, resulting in the phenomena of concept drift, which is one of the main problems that affects data streams mining. In this study, we present an approach to explore and understand the concept drift occurring in Twitter data streams. Two machine learning technique Naive Bayes Classifier and eXtreme Gradient Boosting (XGBoost) Classifier were applied on more than 11K tweets focused on two technology products (iPhone 13 & iPhone 14), and to detect / understand concept drift and specify whether concept drift in a technology area is a radical or an incremental innovation.

Key Words: Tweets Mining, Social Media Analysis, Concept Drift, Data Stream,

ÖZET

TWEETLERİN METİN MADENCİLİĞİ YOLUYLA TEKNOLOJİDEKİ KAVRAM SAPMASINI KEŞFETME

NACİ, Mohamad
Yüksek Lisans Tezi, Endüstri Mühendisliği
Danışman: Doç. Dr. Alptekin DURMUŞOĞLU
Kasım 2022
59 sayfa

Son on yılda bilgi kaynakları ve dijital çağda kullanımlarında dramatik bir dönüşüm yaşandı. Sosyal medya ağları, bireylerin duygularını ifade etmenin yeni bir yolunu getirdi. Şirketler, resmi kurumlar ve çeşitli kuruluşların iletişim sitelerinde çeşitli gelişmeleri, ürünleri, görüşleri ve hatta bazen resmi kararları paylaştıkları sayfalar olduğu için konu bazı kişilerin ayrı görüşlerinin ifadesi olmaktan çıkmıştır. Sosyal medya, kullanıcıların çeşitli duygu kategorilerinde sınıflandırılabilen diğer kullanıcıların görüşlerine erişmesine izin veren ve karar vermede yavaş yavaş önemli bir rol üstlenen, çok miktarda bilgi içeren bir ortama dönüşmüştür. Twitter, dünyanın herhangi bir yerinde, her an neler olduğunu açıklamak için oluşturulmuş bir mikroblog hizmetidir. Yaklaşık 1,3 milyar insandan günde 500 milyondan fazla mesaj için büyüleyici bir forum. Twitter verileri kısa, spesifik ve kolay erişilebilirdir, bu nedenle veri akışları madenciliği yoluyla duygusal analiz için en iyi kaynaklardan biri haline gelir. Temelde yatan veri dağılımının zaman içinde değişebilmesi ve bunun sonucunda fikir kayması olgusuna yol açabilmesi, veri akışları madenciliğini etkileyen temel sorunlardan biridir. Bu çalışmada, Twitter veri akışlarında meydana gelen kavram kaymasını keşfetmek ve anlamak için bir yaklaşım sunuyoruz. İki teknoloji ürününe (iPhone 13 ve 14) odaklanan 7K'den fazla tweet'e iki makine öğrenimi tekniği olan Naive Bayes Sınıflandırıcı ve eXtreme Gradient Boosting (XGBoost) Sınıflandırıcı uygulandı ve kavram sapmasını algılamak/anlamak ve bir teknolojide kavram kayması olup olmadığını belirlemek için alan radikal veya artımlı bir yeniliktir.

Anahtar Kelimeler: Tweet Madenciliği, Sosyal Medya Analizi, Kavram Kayması, Veri Akışı.

''Dedicated to my family''

ACKNOWLEDGEMENTS

On the accomplishment of this thesis, I feel grateful to all those people who provided me with help and encouragement to complete my research.

Foremost, I'd like to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Alptekin DURMUŞOĞLU for his guidance, patience, and support.

I am deeply grateful to my beloved parents; Ridvan NACİ and Fatima NACİ, for their endless support and all the sacrifices they made for me.

My last but largest thanks go to my beloved wife; Hedil AL-SHIHABI who made it possible to accomplish this thesis by her endless support and love. And I praise Allah, for giving me these wonderful human beings in my mortal life and a chance to be where I want to be most.

Mohamad

TABLE OF CONTENTS

Page

LIST OF TABLES

Page

LIST OF FIGURES

Page

LIST OFABBREVIATIONS

IoT	Internet of Things
5Vs	Volume, Velocity, Variety, Veracity and Value
SA	Sentiment Analysis
OM	Opinion Mining
XGBoost	Extreme Gradient Boosting
SVM	Support Vector Machines
K-NN	K-Nearest Neighbor
NB	Naïve Bayes
DDM	Drift Detection Method
EDDM	Early Drift Detection Method
LLD	Learning with Local Drift Detection
FW-DDM	Fuzzy Windowing Drift Detection Method
LFR	Rate drift detection
DDE	Drift Detection Ensemble
API	Application Programming Interface
IDE	Integrated Development Environment
NLP	Natural Language Processing
BoW	Bag of Words

CHAPTER I

INTRODUCTION

1.1 Preface

The acceleration in the volume of data produced daily is growing constantly due to the increased number of applications that generate massive amounts of data at a great velocity, such as the content and reactions on social media, advertisements click, shares, transactions, streaming content, the Internet of Things (IoT) realms and so much more. This exponential growth of data (that may form valuable source of information) is called **big data**, which may primary generated mainly through the following source's types:

- *Commercial data* this data can be purchased from companies, organizations, or specialized social media.
- *Social data* that comes from social media sources such as Instagram, YouTube, Facebook, Twitter, ...etc. This kind is recently growing massively in a short time.
- *Public data* like weather, economic, and socio-economic data (Figure 1).
- *Operational data* comes from transactional systems, this data is coming from sensor networks (monitoring and streaming data).
- *Dark data* this kind is usually owned by users but not used for decision support or making, such as contracts, reports and emails. (Kantardzic, 2020)

Table 1.1 Big Data available online

Company	Big Data
YouTube	One hundred hours of videos added every 60 seconds
Facebook	Over 1.4 billion active users
Twitter	175 M tweets every 24 hours
Google	Two million search every 60 seconds
Instagram	Forty Million picture/photo every 60 seconds

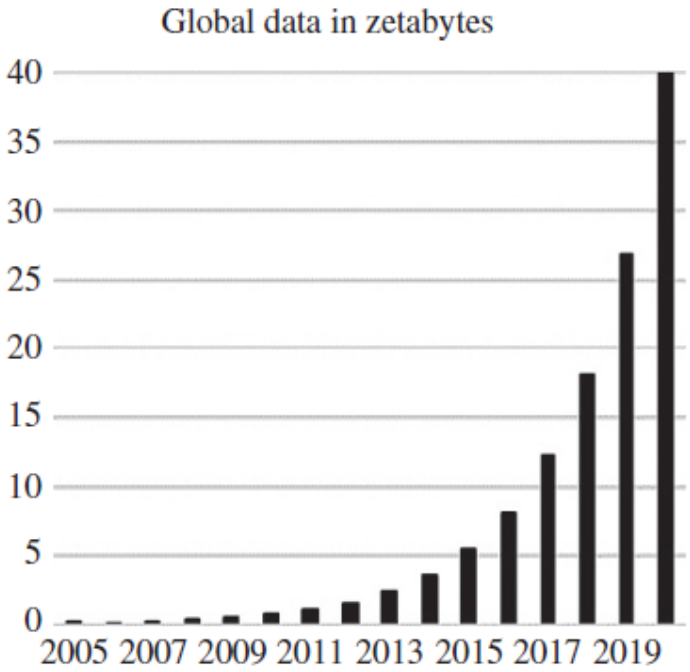

Figure 1.1 Global data growth.

The necessity to cope with massive volumes of data has driven data mining research aimed at generating knowledge from data, which may be interpreted as uncovering new and non-trivial patterns, relations, and trends in data relevant to the user (Schuh, et al., 2019).

Data streams, which is defined by (Aggarwal, 2007) as an ordered sequence of items that arrive in timely order, can be divided to two types: static and dynamic data streams, which has become of more interest recently by researchers because of its use in many domains such as: industrial sensor networks, fraud detection, spam filtering, customer needs prediction, online marketing, and etc. Data streams have five main characteristics that make dealing with them is not an easy mission, which called (5Vs):

1. **Volume**: the size of data, which is increasing rapidly.
2. **Velocity**: the accumulation speed of data.
3. **Variety**: the data can be structured, semi-structured and unstructured due to the variety of data sources.
4. **Veracity**: refers to the accuracy, credibility, and quality of data.
5. **Value**: which means the ability to benefit/ extract from data and depend on it in decision making.

Data streams areas of application are very wide, and they are getting more importance day after day, the following figure shows some of these areas (Agrahari, Singh, 2021).

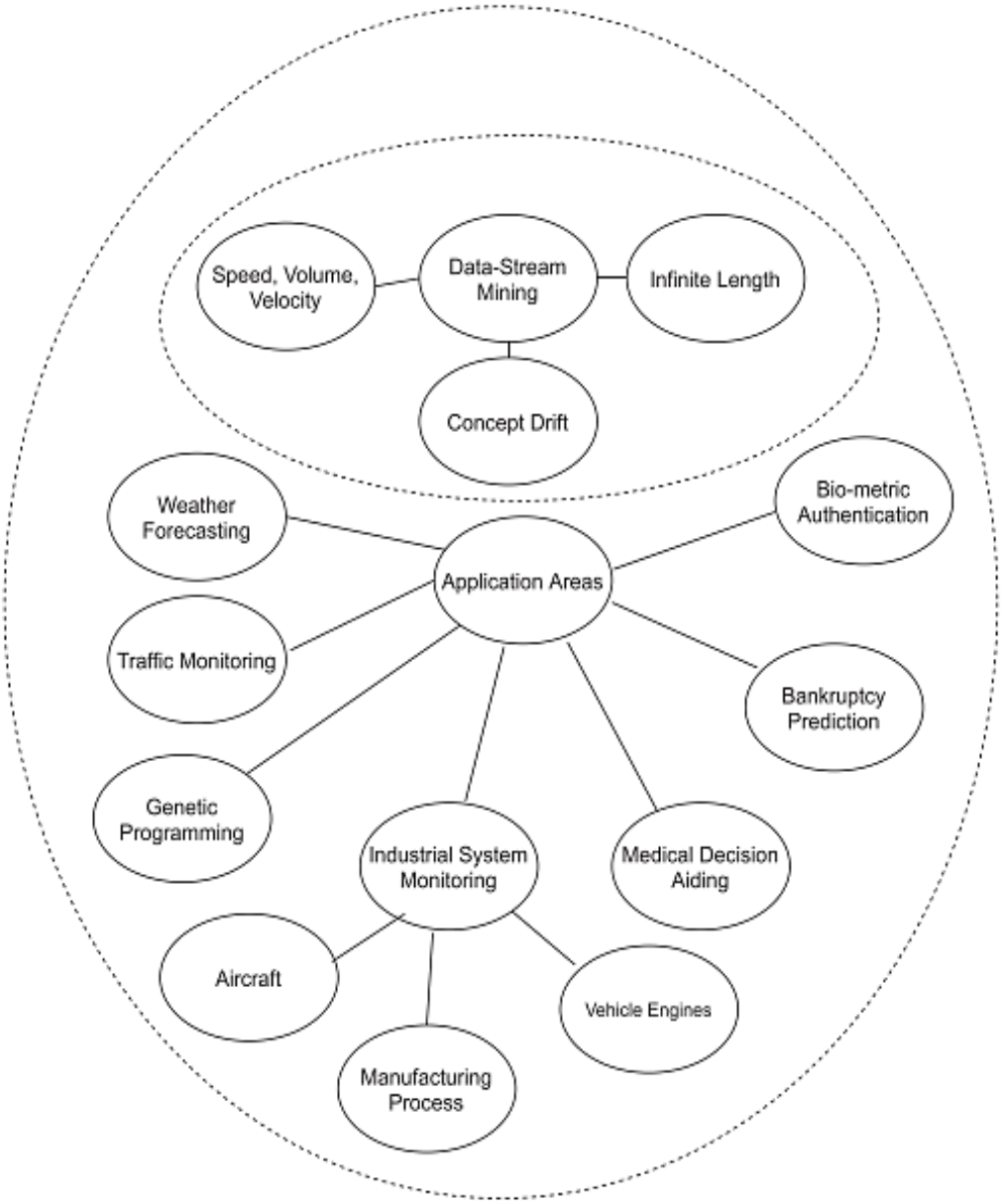

Figure 1.2 Data Streams areas of application and mining

Over the last decade, a dramatic transformation has happened in data sources and their use in the digital era. Social media networks have brought a new way of expressing the sentiments of individuals. The matter went beyond being an expression of separate opinions of some individuals as companies, official institutions and various organizations have pages on the communication sites through which they share various developments, products, opinions, and sometimes even official decisions. Social media has developed into a platform with a wealth of information where users may

read other users' opinions, which are divided into different sentiment groups and are becoming increasingly important in decision-making. Twitter is a microblogging service built to describe what is happening anywhere worldwide, at any moment. Twitter data is short, specific, and easily accessible, that's why it has become one of the best sources for sentimental analysis and knowledge discovery by stream mining. Sentiment analysis (SA) or opinion mining (OM) is the study of public opinions, sentiments, emotions, and attitudes expressed in social media (Bhuvaneswari, Srividhya, 2017).

One of the major issues that will be faced when dealing with data streams is concept drift, which is the change in data distribution over time that happens in evolving/non-stationary environments (Widmer, Kubat, 1996). Concept drift between time point *(t)* and time point *(t+1)* can be expressed as:

$$P_{t+1}(x, \omega) \neq P_t(x, \omega) \quad (1.1)$$

Where ω represent the class (concept) and x represents the data variable, P_t and P_{t+1} are the joint probability distributions at time *t* and *t*+1, (Ditzler, Polikar, 2013)

Sentiment analysis is a complex procedure that involves many steps, including subjectivity analysis, and sentiment orientation. It is regarded as a brand-new, developing area of machine learning research (Saberi, Saad, Sentiment Analysis or Opinion Mining: A Review, 2017). The fact that the underlying distribution of data may vary over time, resulting in the phenomena of concept-drift, is one of the main problems that affects data streams mining. The comprehensive study of locating and managing concept-drift in this stream of developing data is known as concept-drift analysis (Vijayalakshmi, 2015). When dealing with concept drift, we may either toss out the outdated information and retrain the model using fresh data (sudden drift), mix the outdated information with the fresh data to gradually update the model (gradual drift), or leave the model alone (no drift) (Yang, McClean, Donnelly, Burke, Khan, 2022).

1.2 Objective and Motivation:

Although there are many studies on the literature that have focused on defining and detection concept drift in data streams, but the law number of studies that focus on understanding the effects of this issue on decision making was our motivation to proceed in this field. Moreover, social media analysis using machine learning tools

and approaches is and emerging area of research recently because its applications can be found in business intelligence, international politics, community studies, economic analysis, and many other fields. This study aims at exploring and understanding the concept drift occurring in Twitter data streams of international technology companies over specific period (almost 10 years), by applying two machine learning technique Naive Bayes Classifier and XGBoost Classifier, trying to answer the following two research questions:

- **RQ1:** Is concept drift in a technology area a radical or an incremental innovation?
- **RQ2:** Can we understand/explore a technological change (innovation) by analyzing tweets?

To the best of our knowledge, this is the first study that focuses on understanding and explaining the effect of concepts drift in technology by tweets mining depending on sentiment analysis (SA). The study presents a new understanding of the changes caused by concept drift on future directions of technological developments.

1.3 Thesis Organizing:

The thesis is organized as follows; Chapter 2 presents fundamentals about Sentiment Analysis of Twitter data and Concept drift's definition, types, and detection methods, and spots the light on the previous work in the literature about these topics. Then, the work methodology with explanation of each od work phases in addition to the results of it was introduced in Chapter 3. Python programming language was used in experimental work of this study, where two machine learning technique Naive Bayes Classifier and eXtreme Gradient Boosting (XGBoost) Classifier were applied on more than 11K tweets focused on two technology products (iPhone 13 & iPhone 14) as a source of data. Chapter 4 included a discussion of the results presented in the previous chapter, conclusion, and the future work.

CHAPTER II

BACKGROUND AND RELATED WORK

Previous work on sentiment analysis and concept drift definition, types and detection methods will be presented.

2.1 Data Mining:

Over the last decade, all fields of science, business and engineering needed to understand/deploy or deal with huge, complex, and information-rich data sets. In business, customer data is becoming a strategic asset and main source for knowledge-based decision making within the highly competitive world that we are living in. The entire operations/process of applying computer-based methods and techniques for knowledge discovery from data sets is known as data mining.

Practically, the primary aims of data mining are prediction and description; in *prediction*, some data variables in the data sets are being used to predict unknown variable or future values of known ones. Under prediction, the aim of data mining is to create a model (executable code) to be used for prediction, classification estimation or similar tasks.

On the other side, *description* involves finding hidden patterns that describes the data set which may be used/interpreted by human, so the goal here is to form an understanding of the studied system by finding the hidden/undiscovered patterns and/or relationships within big data sets.

Based on it, the data mining activities can be classified as follows:

A. Predictive: through which we produce the model for the system of given data set, or
B. Descriptive: in which new information is produced based on a known data set.

The key data mining tasks are:

1. **Classification**: Finding a predictive learning function to classify the data elements

into one of the already defined classes

2. **Regression**: Mapping data item(s) to real value(s) prediction variable(s)
3. **Change and deviation detection**: Discovering the most significant changes in the data set.
4. **Clustering**: Identifying set of categories or clusters to describe the data.
5. **Dependency modeling**: To create a model that can show the variables' dependencies or the features' values within given data set or part of it.
6. **Summarization**: Finding a way to describe a data set in a compact form. (Kantardzic, 2020)

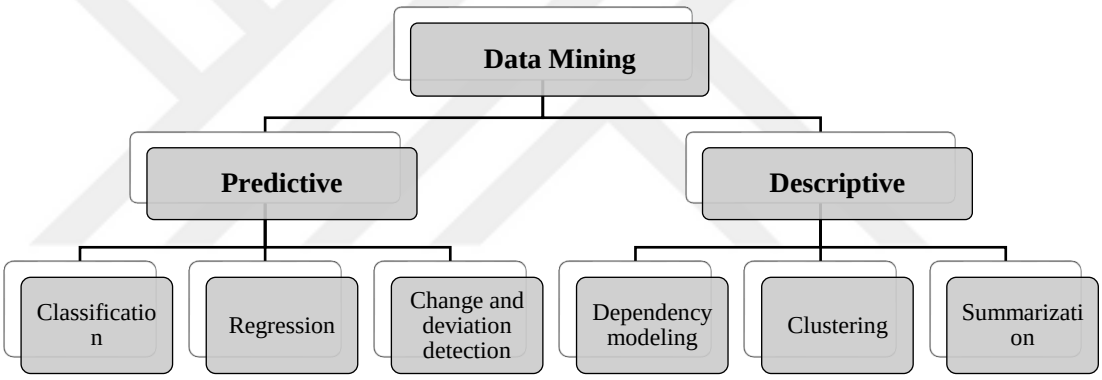

Figure 2.1 Key Tasks of Data Mining

2.2 Knowledge Discovery Process:

The process of knowledge discovery consists of nine iterative and interactive steps (Figure 2.2). It's also iterative in each step, which means that it's required sometimes to go back and adjust/modify the pervious steps.

The process itself requires deep understanding, in addition to knowing the needs and requirements of each step, Taxonomy for the Data Mining methods can help here, which is presented in the next section. The process begins with specifying the knowledge discovery goals and ends with applying the knowledge discovered/reached.

Following are the nine steps (Maimon, Rokach, 2010):

1. Understanding the domain of application: It's required to define the aims of the

end users and the surrounding environment where the knowledge will be applied.

2. Selecting or creating the data set: Finding out which data is available/applicable for the process.
3. Pre-processing and cleaning: this may include removing noise and handling missing values. It can be done through Data Mining algorithms or complex statistical methods.

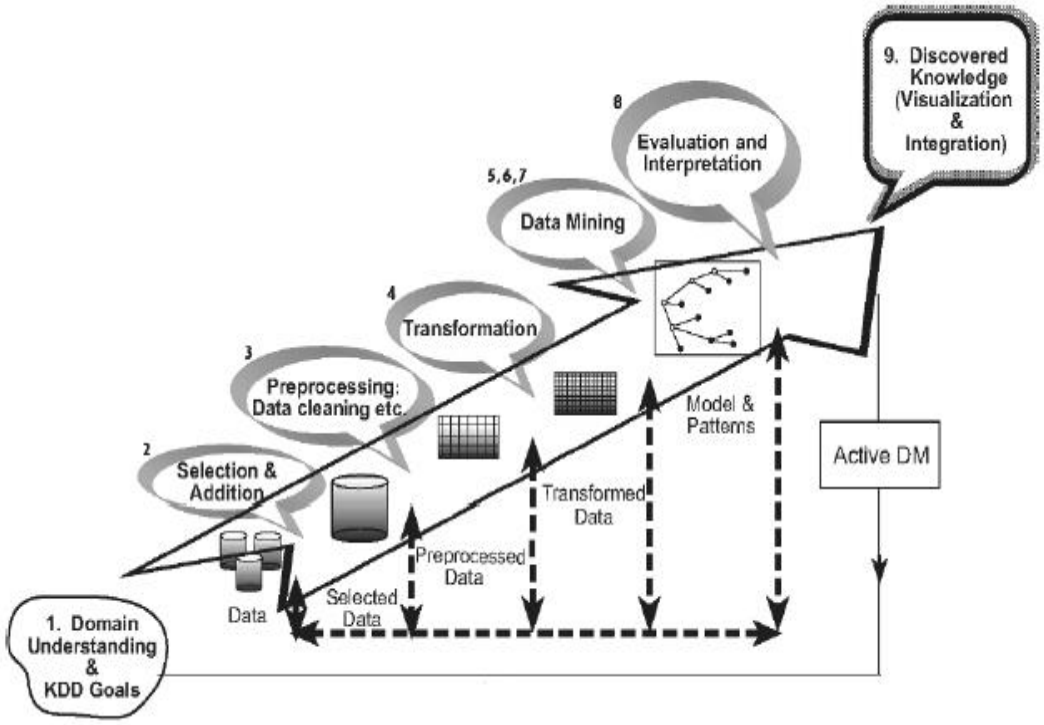

Figure 2.2 The Process of Knowledge Discovery

4. Data transformation: which includes generating better data to be used in Data Mining. Methods here include dimension reduction and attribute transformation.
5. Specifying the suitable Data Mining task: It depends on the goals determined in the first step, and on the previous steps.
6. Choosing the algorithm: Each algorithm has its requirements, parameters, and tactics of learning.
7. Employing the algorithm: It may be needed to apply the algorithm several times until the desired result is obtained.
8. Evaluation: Is the model useful and/or comprehensive?
9. Deploying the knowledge: Using the knowledge into other systems for further actions, some changes can be made in the system to measure the effects.

2.3 Taxonomy of Data Mining Methods:

As there are many Data Mining methods that can be used for many purposes, Taxonomy can help to understand the variety of methods, and know more about their interrelation and grouping. It's can help to differentiate between Verification-Oriented Data Mining and Discovery-Oriented one (Figure 2.3) (Maimon, Rokach, 2010).

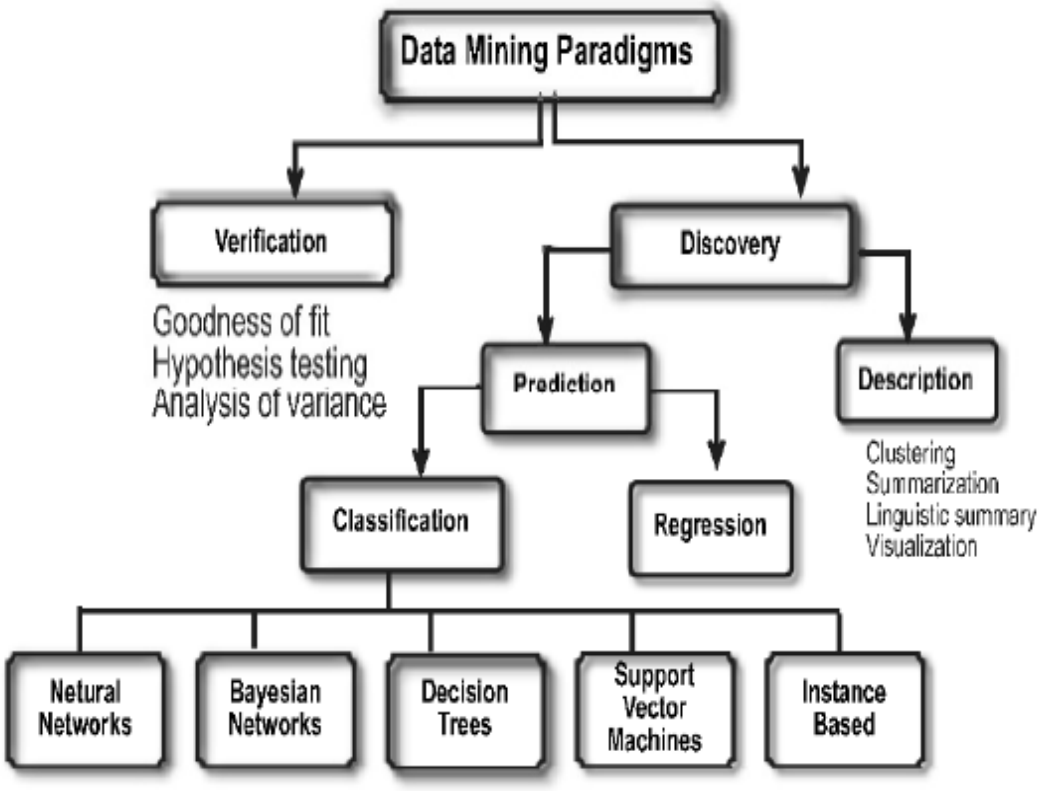

Figure 2.3 Data Mining Taxonomy

We can also refer to the prediction approaches as **"supervised learning"** and on the other hand, **Unsupervised learning** refers to modeling the distribution of instances in a typical, high-dimensional input space. the term **"unsupervised learning"** refers to only a part of the description methods presented in Figure 2.3. For example, "clustering methods are included under it, but visualization methods are not. Unsupervised learning refers mostly to techniques which classify the instances without a dependent, prespecified attribute. While the supervised methods attempt to discover the relation between input attributes (independent variables) and a target attribute (dependent variable).

2.4 Text Mining:

Text mining is a branch of data mining that focuses on finding patterns in textual data that may be important to form and explore knowledge (Sagayam, S.Srinivasan, Roshni, 2012). Different data mining tasks such as: classification, clustering, summarization, etc. can be applied on textual data sets to extract knowledge. Text mining intersects with many fields such as data mining, machine learning, computational linguistics, information retrieval and statistics (Figure 2.4) (Weiss, Indurkhya, Zhang, Damerau, 2005).

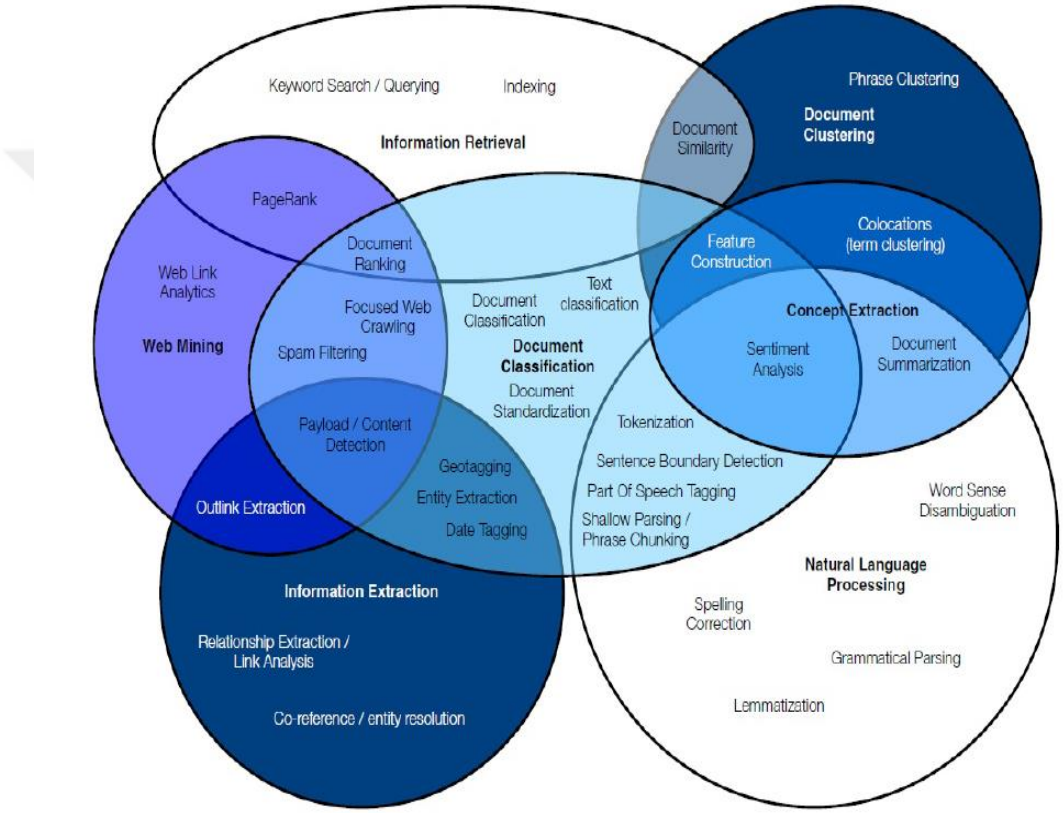

Figure 2.4 Inter-relationship among different text mining techniques and their core functionality

Text data sources are massively expanding, especially with the wide spread of social media, making huge amounts of data available. Text mining functions or tasks such as sentiment analysis, opinion mining, predictive & trend analysis and feature extraction can be applied in academia, industry, marketing, business analysis, social studies, and many other fields.

2.5 Sentiment Analysis:

Sentiment Analysis (SA) -or opinion mining (OM)- is a branch of text mining. It is a widely used text classification tool that analyzes the source text and discovers the underlying sentiment (predicting people's feelings or emotions about something) which can be either positive, negative, or neutral. The analysis can be done from different directions such as Natural Language Processing methods, application of lexicons with annotated word polarities, along with some machine learning-based approaches.

2.5.1 Sentiment Analysis Approaches:

There are three widespread groups for sentiment analysis:

A. Machine Learning Approach: Depends on machine learning techniques to give information about the polarity of sentiments. To perform classification, two collection of text document are needed: training collection and test collection; the first one is used by the classifier to differentiate between text features, and the second one is used to discover the accuracy of classification/prediction. There are many machine learning algorithms used for sentiment analysis, however the performances of Support Vector Machines, Naive Bayes, and Maximum Entropy with SA and classification are highly successful (Saberi, Saad, Sentiment Analysis or Opinion Mining: A Review, 2017). Other approaches include K-Nearest Neighbor, Random Forests, and XGBoost. Many researchers have compared these approaches on text data to find the best classifier in sentiment analysis tasks, Agarwal et al. found that Naïve Bayes classifier obtained good results compared to Support Vector Machine on reviews in Cantonese (Agarwal, Xie, Vovsha, Rambow, Passonneau, 2011). On the other hand, many researchers found that Support Vector Machine is the best classifier for text classification (Saberi, Saad, Sentiment Analysis or Opinion Mining: A Review, 2017). Sabri and Saad have presented an overview of sentiment analysis approaches, numerous methods utilized in this subject, along with the application domains and difficulties for sentiment analysis. The research concluded that further research, particularly in languages other than English, is still needed before social media or networking sites can be used as a source of data for SA or OM purposes.

Machine Learning approaches are usually classified as:

1. **Supervised Learning:** requires well-defined and well-labelled corpus as a training set for the model, and another data set to test the model performance & accuracy, many ML algorithms go under this category such as: Support Vector Machine, Random Forests, XGBoost, Naïve Bayes and other classifiers.
2. **Unsupervised Learning:** in this method, no training data set is needed, only input data set is required, and unsupervised methods can be either machine learning based, or lexicon based, an example of these methods is clustering, in which semantic orientation approach is used and algorithms will extract the phrases that include adjectives or adverbs to predict the sentiment orientation.
3. **Semi-Supervised Learning:** this can be considered as a middle solution between the two previous ones, where a series of labelled and unlabeled data is provided, for the purpose of classification.

B. Lexicon-Based Approach: Depending on an unsupervised learning technique since no training is required under this approach. It can be said that this method determines whether the term is far or close to being positive or negative, that is being done depending on lexical rules. Some researchers introduced a sentimental lexicon (Abbasi, Chen, Salem, 2008), the authors argued that that the preparation of manual lexicon is higher effective than the preparation of an automatic sentiment lexicon. Under Lexicon-Based approach there are two categories:

1. **Dictionary-based approach:** in which small set of known-orientation words are collected manually, then this set is increased by searching of well-known corpora for their antonyms and synonyms (Hu, Liu, 2004).
2. **Corpus-based approach**: this method depends on syntactic patterns which comes together along with seed list of opinion words, aiming to discover other opinion words within big corpus, this method is not effective compared with the previous one because it's difficult to come up with a large corpus consisting of all words of the studied language.

C. Hybrid Approach: lexicon-based machine learning approach that includes manual-written linguistic rules. In this method, cascaded classifiers are used so if one of them failed, the next one performs the classification task, and so on till the categorization of text/document is finished.

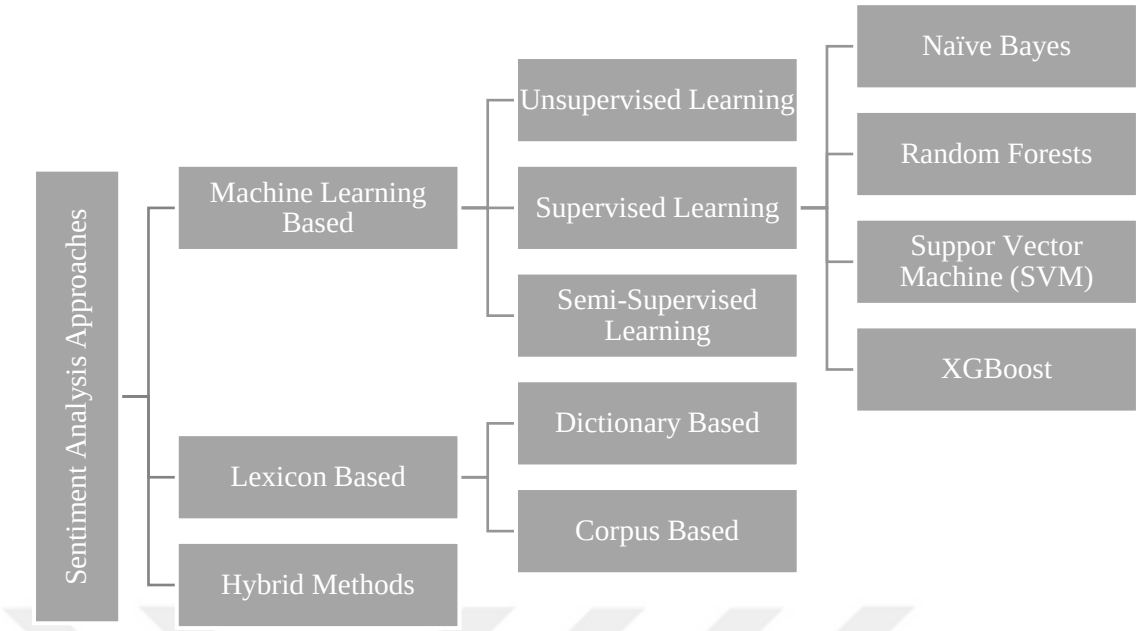

Figure 2.5 Sentiment Analysis Approaches

2.5.2 Sentiment Analysis of Twitter Data:

Social media is exceptionally useful source which generates huge amount of sentiment rich data including tweets, online blogs, status updates…etc.

Twitter is a microblogging service built to describe what is happening anywhere worldwide, at any moment. It is a fascinating forum for more than five hundred million messages per day from about 1.3 billion people. Twitter data is short, specific, and easily accessible, that's why it has become one of the best sources for sentimental analysis and knowledge discovery by data streams mining.

Sentiment analysis of Twitter data is challenging work due to the use of slang, misspells, abbreviations and the unstructured nature of data which required a lot of preparation and cleaning work on data set to be ready for machine learning approaches.

The Naive Bayes technique was used in (Gangawane, Torvi, 2017) to assess sentiments on abbreviations and short sentences, to improve the algorithm for identifying sentiment words, and to successfully handle bipolar sentiments based on Twitter data.

Many of works related to sentiment analysis have focused on product or movies reviews (Devi, Bai, Ramasub, 2020), (Ogul, Ekmekciler, 2012), (Wang, Liu, 2017),

(Dave, Lawrence, Pennock, 2003) and (Alencia, Nizar, Ayuning, Herkules, 2018) on customer tweets, review sites, blogs or other websites.

(Zhou, Tao, Yong, Yang, 2013) have suggested a Tweets Sentiment Analysis Model (TSAM) that may identify societal interest in and attitudes on the 2010 Australian federal election. The research has shown that it is feasible and advantageous to construct an intelligent system for sentiment analysis based on a lexicon.

In (Vasudevan, 2017) the sentiment of twitter data was performed depending on Support Vector Machine (SVM) and Decision Tree (DT), and that was performed depending on a dataset consists of 7156 tweets classified with respect to Google self-driving cars, achieving high accuracy with SVM (90%) and less accuracy with DT (65%). Similar work was introduced in (Barzenji, 2021) Whereas sentiment analysis on Twitter text was used to learn about the subjective polarity of the writings, this analysis was carried out using three distinct machine learning algorithms: Support Vector Machine, Radom Forests, and Gaussian Naive Bayes. The classifiers obtained accuracy of 89%, 88%, and 72%, respectively.

Similar method was employed in (Gupta, Pruthi, Sahu, 2017), where part of that process on sentiment analysis of Twitter data using a mix of two machine learning algorithms, K-Nearest Neighbors (KNN) and Support Vector Machines (SVM). According to the findings, the suggested model enhanced the accuracy and f-measure of tweet class prediction.

2.6 Concept Drift Definition & Types:

Concept drift has become a communal area of research in the field of data mining and machine learning since it's a major issue when dealing with data streams. Some previous works has focused on the definitions and types of concept drifts, such as: Concept drift is defined in (Elwel, Polikar, 2011) as a mutation in the definitions of the classes (concepts) over time, which results in a change in the distributions from which the data for these concepts are derived. Real drift and virtual drift are the two categories into which variations in data distribution are divided (Widmer, Kubat, 1996). These two categories of drifts are formalized as follows (Janardan, 2017) and (Gama, Zliobaite, Bifet, Pechenizkiy, Bouchachia, 2014):

- **Real Drift:** irrespective of changes in the evidence $P_t\ (x)$, the posterior probability

$P_t(x|Y)$ drifts with time (x).

- **Virtual Drift:** occurs when the evidence or the data's marginal distribution, $P_t\ (x)$, changes without altering the posterior probability of classes, $P_t(y|X)$

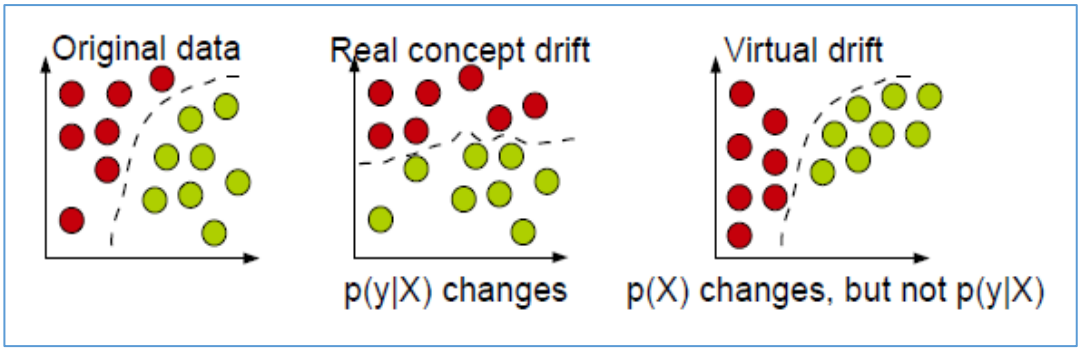

Figure 2.6 Types of drifts: circles represent instances, different colors represent different classes

Another categorization of concept drift is based on how the drift evolves in the system, The drift may occur suddenly, incrementally, or gradually. It used the phrase "intermediate concept" to represent the shift between concepts to better demonstrate the aforementioned categories (Gama, Zliobaite, Bifet, Pechenizkiy, Bouchachia, 2014).

The changes in underlying data may occur in different forms.

- The drift can be sudden or abrupt; this happens when there are sudden changes in the concept of the model.
- The drift can be incremental when the old concept incrementally changes to a new concept over a period.
- The drift may be gradual; it means that the change is not abrupt, and it takes a long time to happen.
- The drift can be recurring drifts when new concepts, or previously seen concepts may repeat after some time.

Figure 2.7 illustrates this classification (Lu, et al., 2019):

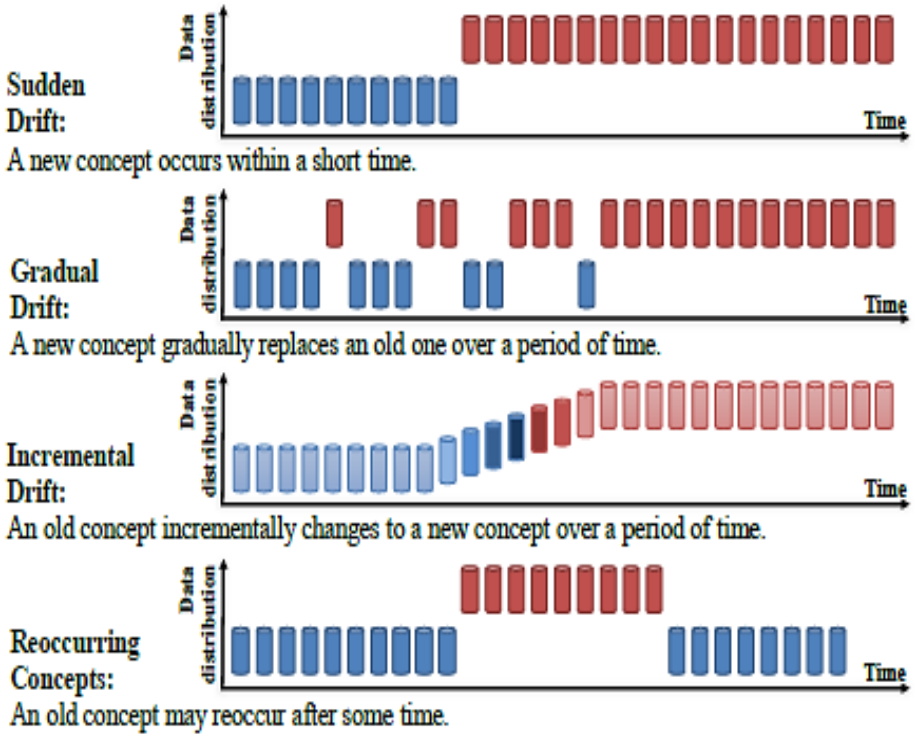

Figure 2.7 Concept Drift Types

2.7 Concept Drift Detection:

The number of surveys, overviews, and reviews on idea drift detection that were published between 2009 and 2019 was provided by the authors in (Gemaq, Costa, Giusti, Santos, 2020)., the result is shown in Figure 2.8:

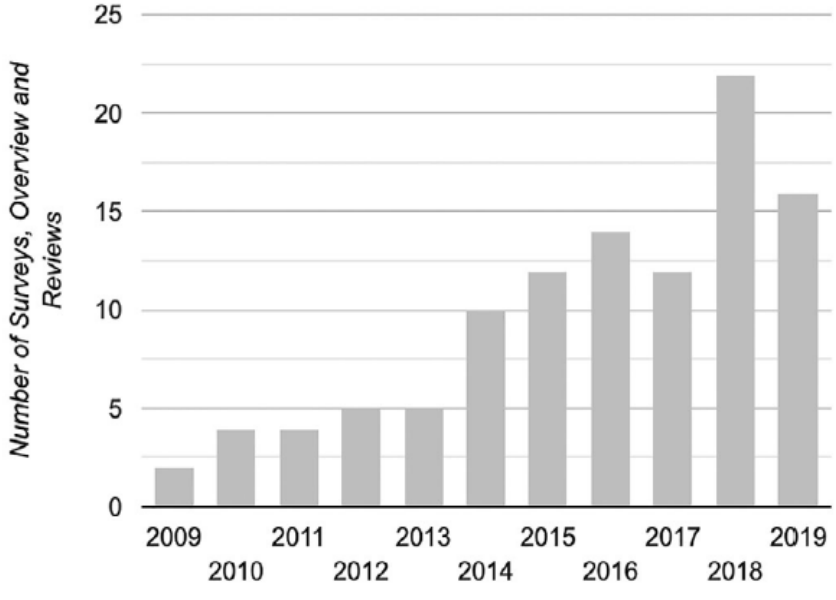

Figure 2.8 Number of publications on concept drift detectors

Figure 2.9 (Bayram, Ahmed, Kassler, 2022) depicts a generic strategy for detecting concept drift. The null hypothesis in the figure is that the test statistic will not show a significant difference between the old and new data, implying that no concept drift was identified. If the null hypothesis is not rejected, the system will stick with the current learner and move along the data stream.

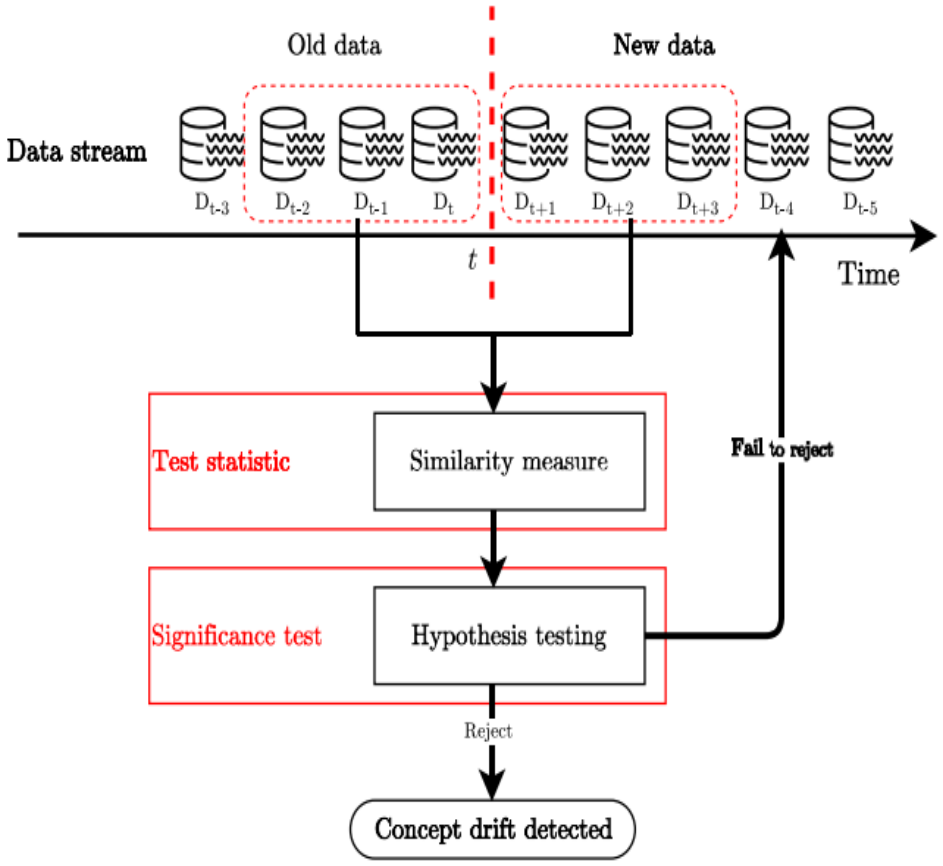

Figure 2.9 A framework of Concept drift detection

Drift detection algorithms can be divided into three categories (Lu, et al., 2019):

2.7.1 Error rate-based drift detection:

The algorithm will monitor changes in the online error rate and will trigger the drift alarm when a substantial change (increase or reduction) of the error is established. Examples of this type of algorithm are:

A. **Drift Detection Method (DDM)** (Gama, Medas, Castillo, Rodrigues, 2004), which uses a binomial distribution. That distribution gives the general form of the probability for the random variable that represents the number of errors in a sample of n examples. The concept is modeling the error as a binomial variable, which means that the expected value of the errors can be calculated.

DDM monitors the count of errors that resulted from a model learned on the previous stream elements, in general, the model error should stay stable or decrease as more data is used assuming that:

a. The learning algorithm/method is controlling overfitting.
b. The distribution of data labels is stationary.

When DDM observes that the prediction error increases, will take is as an evidence of change occurrence.

For better understanding, let P_t the prediction error rate at time t. Since the number of errors within a sample of t examples is modeled by binomial distribution, then the standard deviation at time t will be: $S_t = \sqrt{P_t(1-P_t)/t}$. DDM will store the minimum value of the error rates P_{min}that was noticed during the period t, and the related standard deviation S_{min} of that point, then the algorithm performs the following tests:

- If $P_t + S_t \geq P_{min} + 2 \cdot S_{min}$, the algorithm will declare a warning, and from this point on, the new examples will be stored in anticipation of change declaration.
- If $P_t + S_t \geq P_{min} + 3 \cdot S_{min}$, the change will be declared, and a new model will be built by the examples stored since the warning occurred, replacing the model induced by the learning method which will be discarded.

  This approach has the disadvantage of being too slow in responding to changes, since all examples after the last change will be deployed to compute P_t, which means that P_t is significantly larger than P_{min} because it required a lot of observations after the change, and for slow changes, too many examples will be stored in the memory after the warning, which increases the risk of sample storage overflowing.

  Other than this drawback, the approach is simple and generic, that made it popularly used, and indeed, it can present good performance with gradual changes (when they are not very slow) and abrupt changes (incremental and sudden drifts).

B. <u>Early Drift Detection Method (EDDM)</u> (Baena-Garcıa, Campo-Avila, R. Fidalgo, Gavalda, Morales-Bueno, 2006), which was developed to improve the detection in presence of gradual concept drift, and keeps performing well with sudden or abrupt drift detection. The principle behind this method is to focus on the distance between two successive errors' classification instead of focusing only on the number of errors. During the learning process, because the model is

developing, the prediction will be improved, which means that the distance between two successive errors will be increasing. Let P_i' the average distance between two successive errors, and S_i' its standard deviation, what stored here, is the values of P_i' and S_i' when $P_i' + 2.S_i'$ reaches the maximum value (we call them then P'_{max} and S'_{max}). In other words, the value of $P'_{max} + 2.S'_{max}$ will represent the point in which the distribution of distances between errors is maximum. The algorithm also defines two thresholds:

- $(P_i' + 2.S_i')/(P'_{max} + 2.S'_{max}) < \alpha$ as a warning level, from this point on, the new examples will be stored in anticipation of change declaration.
- $(P_i' + 2.S_i')/(P'_{max} + 2.S'_{max}) < \beta$ as a drift level, which after it, the drift will be declared, and a new model will be built by the examples stored since the warning occurred. Then the values of P'_{max} and S'_{max} will be reset too.

The stored examples will be removed when the similarity between the actual value of $P_i' + 2.S_i'$ and $P'_{max} + 2.S'_{max}$ increase over the warning threshold, and then the algorithm will return to the normal condition.

C. <u>Adaptive Windowing (ADWIN):</u> (Bifet, Gavaldà, 2007)

This is one of the popular methods that goes under the window-based detectors, which in general depends on dividing the data streams in a sliding manner, based on either data size or time interval, then the performance of the latest observations will be compared with a reference window. ADWIN uses exponential histograms to detect or estimate the change, it maintains a variable-length window of recent items, ensuring that the data distribution has not changed. This window then is separated into two smaller windows (W_0, W_1) which are utilized to detect the change. The algorithm will compare the averages of both sub-windows to check whether they correspond to the same distribution. The drift will be flagged if the distribution equality changes. If the drift was detected, W_0 will be replaced by W_1 and new W_1 will be set. To assess whether the two sub-windows represent the same distribution, ADWIN utilizes a significance value $\delta = \in (0,1)$.

In addition to the methods presented above, there are many other methods that depends on the error rate to detect the drift, such as the one class drifts detection (OCDD) (Gözüaçık, Can, 2021) and Page-Hinckley Test (PHT) based drift detector (Qahtan, Alharbi, Wang, Zhang, 2015).

The following table summarize the error-based classification algorithms (Bayram, Ahmed, Kassler, 2022):

Table 2 Error-based drift detectors

Method	Calculation Method	Tested Hypothesis	Type of drift detected
DDM	Online error rate	Distribution estimation	Sudden drift
EDDM	Online error rate	Distribution estimation	Gradual / Sudden
ADWIM	Error rate difference	Hoeffding bound	Gradual / Sudden
Page-Hinckley	Average value	Performance means	Sudden drift
OCDD	Outlier percentage	Post hoc Neymenvi test	Gradual / Sudden

2.7.2 Data Distribution-based Drift Detection:

Techniques in this category use a metric/distance function to quantify the dissimilarity between historical and current data distributions. Since raw data points are used directly rather than through indirect, abstract information (the learner's output or parameters), data distribution-based drift detection approaches have the advantage of sensitive detection and important output knowledge (when, how, and where concept drift happens).

Figure 2.10 illustrates the three components of a general framework for concept drift detection based on data distribution: data modeling, divergence measurement, and statistical significance test. Generally, the data of real-time application doesn't follow any particular distribution, that's why, it is unrealistic to derive an unknown non-parametric distribution F_t straight from the raw data points W_t, The majority of drift detection techniques estimate the non-parametric empirical distribution $\hat{F}_t$ through data modeling.

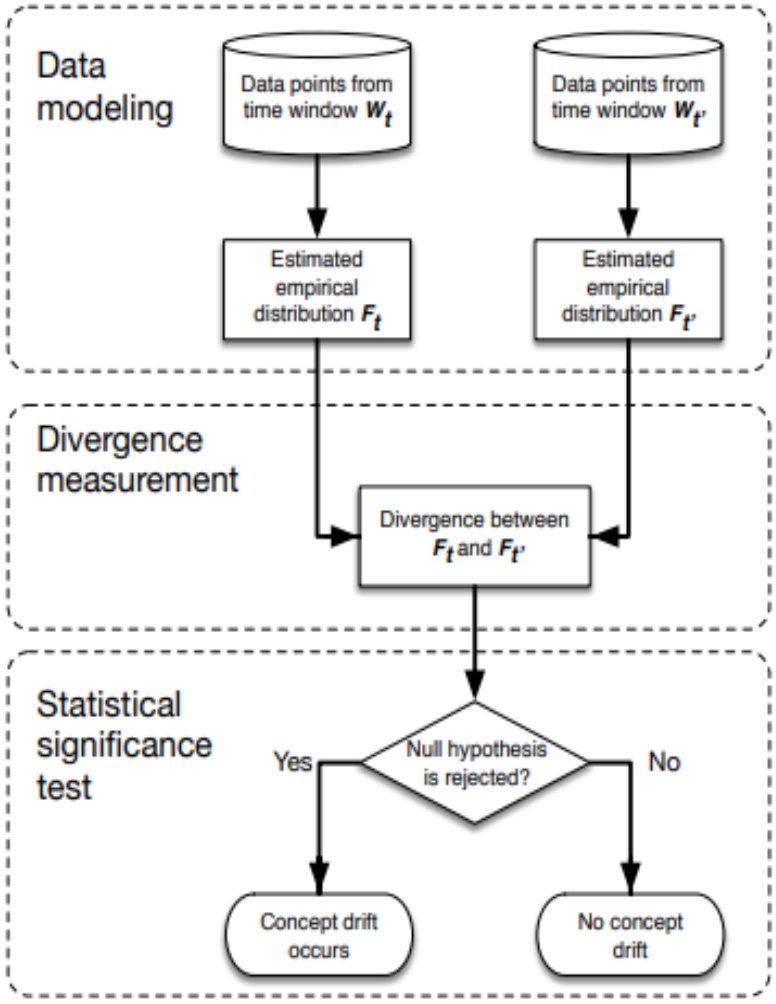

Figure 2.10 Data distribution-based concept drift detection framework

An example for this kind is:

A. Competence Model-based drift detection (CM) (Lu, Guangquan Zhang, Lu, 2007)

This method is an innovative technique for identifying concept drift in a case-based reasoning system. It presents a new competence model that discovers differences through changes in competence rather than evaluating the actual case distribution. No prior knowledge of case distribution is necessary for the competence-based idea detection method to work, and it offers statistical assurances on the dependability of the changes found as well as accurate descriptions and quantification of these changes. When compared to other well-known non-parametric approaches, this detection method has the following advantages:

1) it can be easily applied to multi-dimensional data while maintaining similar results

in one-dimensional data; 2) it is more stable and achieves better results, particularly for small samples, because data can share distribution contributions among related competence areas rather than splitting strictly by cutting edges, making it more tolerable to sample bias; and 3) it can describe detected changes by highlighting some competence.

Another example of Data Distribution-based Drift Detection method is Equal Density Estimation (EDE) (Gu, Zhang, Lu, Lin, 2019).

2.7.3 Multiple Hypothesis Test Drift Detection:

these algorithms apply similar techniques of mentioned in the previous two categories, but they use multiple hypothesis tests to detect concept drift, such as the Linear Four Rate drift detection (LFR) (Wang, Abraham, 2015) and Drift Detection Ensemble (DDE) (Maciel, Santos, Barros, 2015).

Although the different methods and approaches for concept drift detection, but (Agrahari, Singh, 2021) have introduced a general block diagram of concept drift detection shown in Figure 2.11.

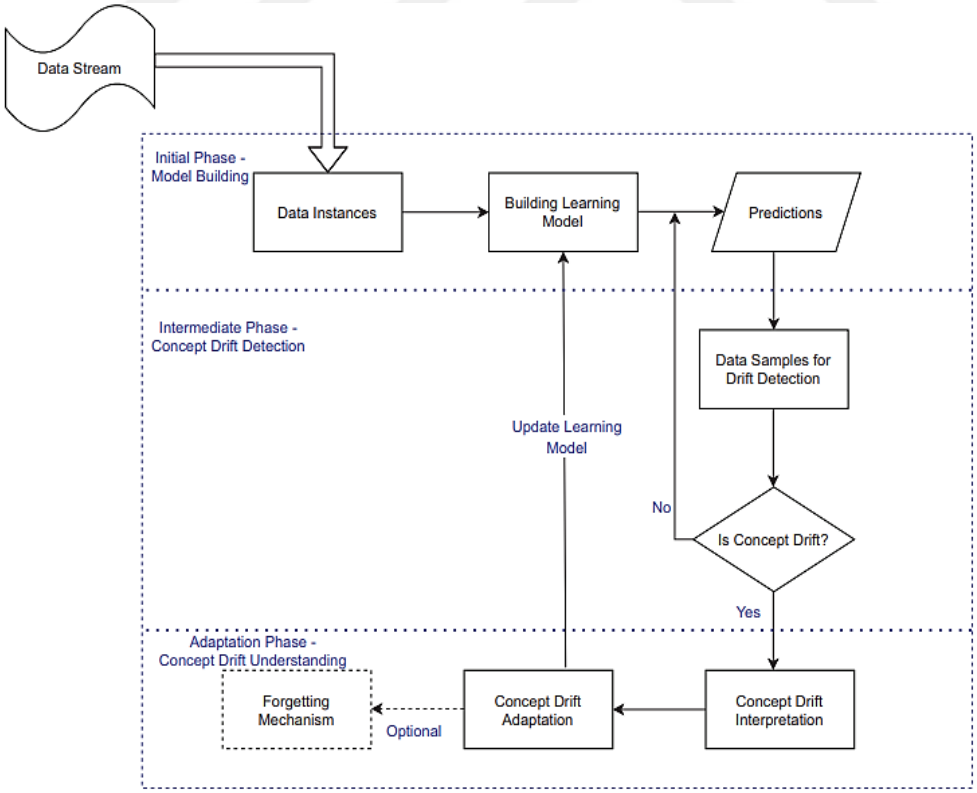

Figure 2.11 Block Diagram for Drift Detection

CHAPTER III
METHODOLOGY

This chapter describes the research processes for collecting and processing twitter data to be evaluated and prepared for machine learning algorithms applied in the following stage.

3.1 Data Source:

Twitter was chosen as source of data due to the following reasons:

- Unlike other social platforms, on Twitter almost all user's tweets are completely public, and
- Twitter supplies a streaming API (application programming interface) that allows developers and academics to retrieve real-time data as well as historical tweets and perform complicated queries.
- Twitter data is specific, since Twitter API gives the ability to get the tweets related to specific topic or pull the non-retweeted tweets of certain user.

Twitter's Application Programming Interface (API) is a platform that allows academics and developers to access both real-time and historical tweet data. APIs can be used to locate and obtain, interact with, or generate a wide range of resources, including (tweets, users, trends, media, and others), as well as to establish the proper period. In this investigation, the Twitter API V2 was utilized to obtain about 11k English tweets Date and Tweet.

The targeted tweets focused on two technology products (iPhone 13 & iPhone 14). The tweets were selected based on two criteria:

I. English tweets.

II. Include one of the keywords iPhone 13 or iPhone 14

The sample is as following:

- 7470 Tweets used to train and test both models, they are in the period between 7/Jan/2020 to 24/Sep/2021.

- 4041 Tweets used to concept drift detection, they are from 24/Sep/2021 to 11/ /Dec/2022.

The data retrieved by Twitter API was pre-processed to remove stop words and punctuations, then stemming/normalization will be applied on tweets to converts the words to their root form (known as lemma). The next stage is sentiment analysis of tweets and splitting dataset to train and test samples with 80/20 ratio so it can be used to train & test the model of Multinomial Naive Bayes Classifier.

Finally, to detect concept drift, a new set of tweets (4K from 24 Sep 2021 to 11 Dec. 2022) will be fetched to test the Naive Bayes model. Finally, to verify the results, another algorithm known as XGBoost will be applied, and the results will be discussed in the results section. Figure 4 presents an overview of the work phases.

3.2 Work phases:

The experimental work was performed according to the following steps:

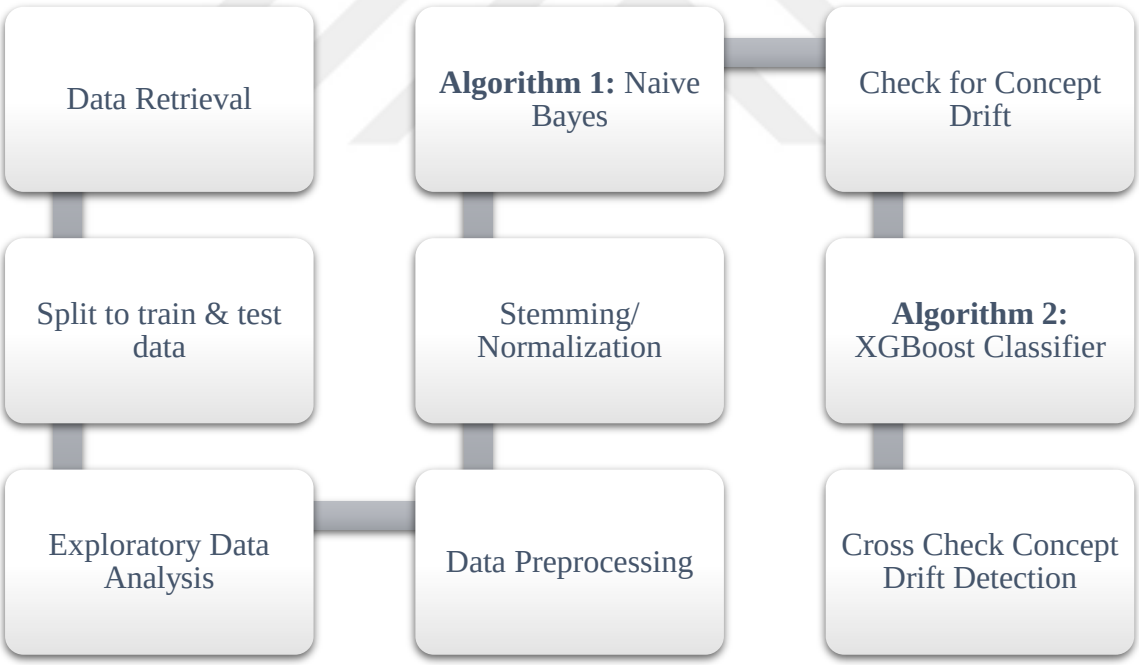

Figure 3.1 Overview of Work Phases

3.3 Data Retrieval

The first timeline of tweets fetched is from 7/Jan/2020 to 24/Sep/2021, Figure 3.2 shows the initial fetched data.

	Date	Tweet
0	2020-01-07	#iPhone13 fact. it may come with no headphone,...
1	2020-01-07	The news that the latest iPhone, the iPhone 12...
2	2020-02-06	iPhone 13 supposedly going to have new camera ...
3	2020-02-08	Gotta skip the next iPhone cause we can't fuxk...
4	2020-02-12	Got some new information regarding iPhone 13.\...
...	...	...
12268	2022-12-10	When we'll get 5G support in iPhone 13...@nawabn...
12269	2022-12-11	I went to the Apple Store yesterday and confir...
12270	2022-12-11	Anyone facing lag in iphone after 16.1.1 updat...
12271	2022-12-11	My battery lasted 2 DAYS without a recharge wi...
12272	2022-12-11	iPhone13\n#キャンペーン \n#一括 #テルル #埼玉 #行田 #前橋 #iP...

12273 rows × 2 columns

Figure 3.2 The Initial Data

The total tweets fetched are divided into two segments to study concept drift. One segment will be used to train and test the models, and the other segment will be used to check for concept drift.

The Python programming language was used in this study, the version used is Python 3.10.6.

The Integrated Development Environment (IDE) used for code writing was Jupyter Notebook, which is an easy-to-use, interactive data science environment across many programming languages.

3.4 Data Pre-Processing:

Benefitting from the streaming API tool, retweets -the republication or forwarding of an existing tweet- are omitted from the original Twitter data set, which lowers the number of duplicated tweets. The initial Twitter data set has already been filtered to include only English tweets.

To facilitate the work, the data is loaded to Pandas data frame which is a python library that makes working with tabular data an easy task in Jupyter notebook. Along with Pandas, various other libraries are used such as:

- 're' which is regular expressions for text preprocessing
- 'NLTK' which helps in Stemming/Normalization, removing stop words etc.
- 'matplotlib' and 'seaborn' for plots and graphs
- 'Scikit-learn' for Machine Learning tasks such as classification, evaluation etc.

The tweets pre-processing also includes making all words in lower case letters and removing stop words and punctuations, while 'emojis' were not removed since they are playing a key role in sentiment analysis. Figure 3.3 presents the code piece of data pre-processing.

```
# Text Preprocessing

def data_processing(text):
    text = text.lower()
    text = re.sub(r'\@w+|\#','',text)
    text_tokens = word_tokenize(text)
    filtered_text = [w for w in text_tokens if not w in stop_words]
    return " ".join(filtered_text)
```

Figure 3.3 Data preprocessing

3.5 Stemming/ Normalization:

Stemming or Normalization is one of the effective techniques, which has been adopted in many different applications, such as machine learning, data retrieval, Natural Language Processing (NLP) and text classification. This technique is usually applied to retrieve information by tracking affixed words back into their root, this will increase the classification accuracy as many studies proved, for example in (Rianto, Mutiara, Wibowo, Santosa, 2021), the results showed that the accuracy of Support Vector Machine (SVM) classifier has achieved a score of 0.85 when applying stemming compared with 0.73 without stemming.

There are different algorithms for stemming, but for English, the most common and effective one is the Porter stemming algorithm (or 'Porter stemmer'). Porter Stemmer is also known for its speed and simplicity in our work, we depended on Porter Stemmer which can be coded in Python as shown in Figure 3.4.

```
# Using Porter Stemmer here

stemmer = PorterStemmer()

def stemming(data):
    text = [stemmer.stem(word) for word in data]
    return data
```

```
# Applying Stemming

twt = twt.apply(lambda x: stemming(x))
```

Figure 3.4 Stemming

3.6 Multinomial Naive Bayes:

The Naive Bayes is a supervised learning technique that uses the Bayes theorem (equation 3.1) for probabilistic classification issues. The technique depends on observing the input data of class (C_k), to calculate the belonging probability of attribute values (v_i) to a specific class.

$$P(C_k|V) = P(C_k)\frac{P(V|C_k)}{P(V)} \tag{3.1}$$

The prediction depends on two assumptions:

- Bag of Words assumption: Assuming that position doesn't matter,
- Conditional Independence: Assuming that the feature probabilities $P(v_i, c_j)$ for inputs $(v_1, v_2, \dots, v_n\ |C_k)$ are independent given the class B.

$$P(v_1, \dots, v_i|C_k) = P(v_1|C_k) \cdot P(v_2|C_k) \cdot \dots \cdot P(v_i|C_k) \tag{3.2}$$

The classifier equations can be described as follows (Kononenko, Kukar, 2007):

$$P(V|C_k) = P(v_1 \wedge \ldots \wedge v_a|C_k) = \prod_{i=1}^{a} P(v_i|C_k) \tag{3.3}$$

With a single application of the Bayes rule we get:

$$P(C_k|V) = \frac{P(C_k)}{P(V)} \prod_{i=1}^{a} P(v_i|C_k) \tag{3.4}$$

And if we applied the Bayes rule again:

$$P(v_i|C_k) = P(v_i) \frac{P(C_k|v_i)}{P(C_k)} \tag{3.5}$$

We get:

$$P(C_k|V) = P(C_k) \frac{\prod_{i=1}^{a} P(v_i)}{P(V)} \prod_{i=1}^{a} \frac{P(C_k|v_i)}{P(C_k)} \tag{3.6}$$

The factor $\frac{\prod_{i=1}^{a} P(v_i)}{P(V)}$ is independent of the class, so by omitting it we get the final expression:

$$P(C_k|V) = P(C_k) \prod_{i=1}^{a} \frac{P(C_k|v_i)}{P(C_k)} \tag{3.7}$$

Owing to its simplicity, the Naive Bayesian classifier often terminates well and is extensively used because it routinely outperforms more complicated classification algorithms. When used to big databases, Bayesian classifiers have also demonstrated great accuracy and speed (Han, Kamber, Pei, 2012).

3.7 Extreme Gradient Boosting (XGBoost):

Extreme Gradient Boosting (XGBoost) is a highly effective machine learning method that depends on gradient tree boosting. It is a decision tree-based ensemble algorithm that uses the gradient boosting framework. XGBoost has been found to produce cutting-edge performance on a variety of conventional classification benchmarks (Li, 2010). The effect of XGBoost, a scalable machine learning method for tree boosting, has been extensively acknowledged in a few machine learning and data mining applications (Tianqi Chen, 2016).

Artificial neural networks tend to outperform most machine learning algorithms when involving unstructured data such as text, images, etc. However, for small to medium structured or tabular data, decision trees algorithms are generally the best in class, Figure 3.5 illustrates the evolution of tree-based algorithms.

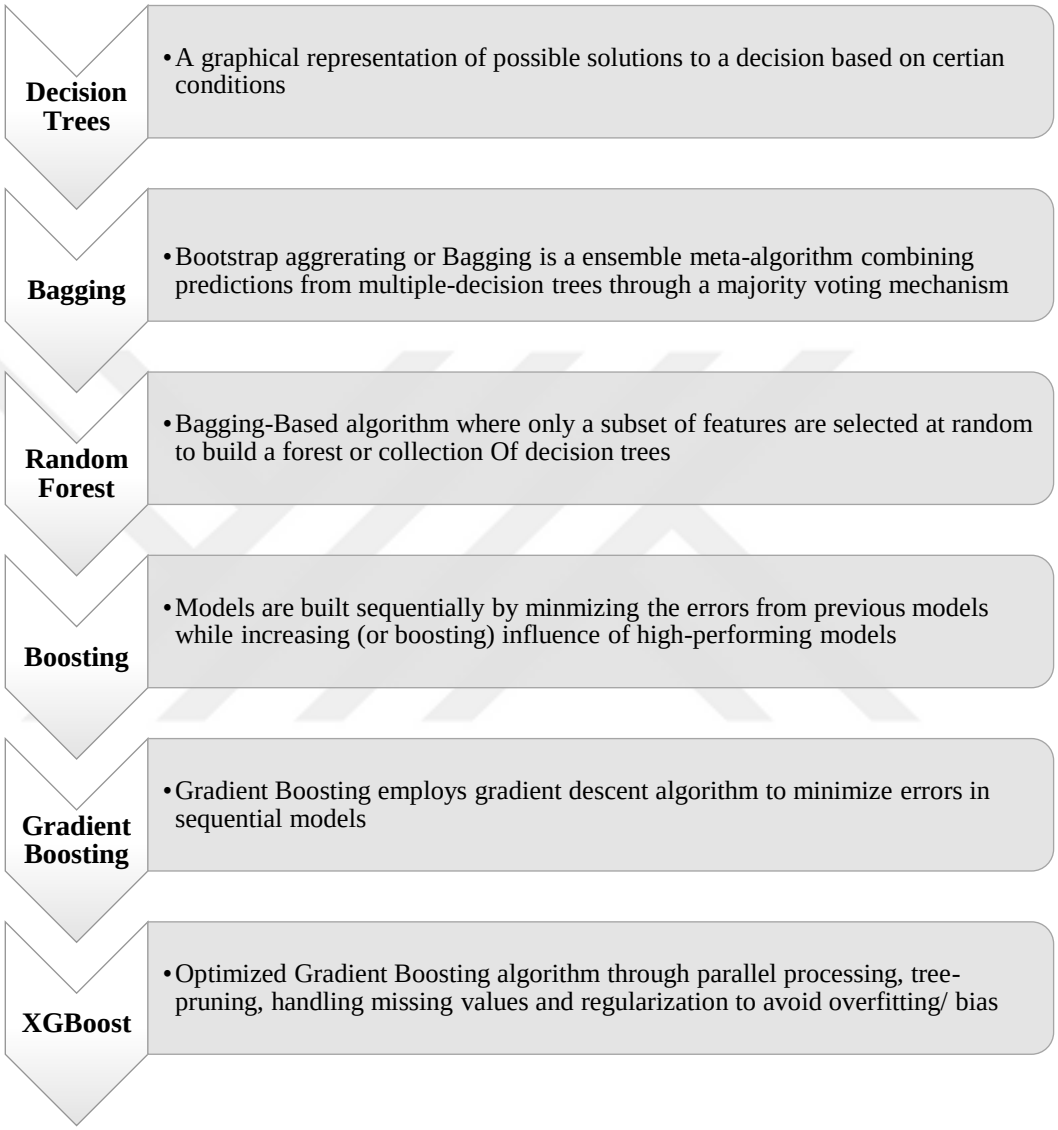

Figure 3.5 Evolution of XGBoost Algorithm from Decision Trees

The base learner of XGBoost is the classification and regression tree (CART), Figure 3.6 shows the structure of XGBoost (Cheng, et al., 2020)

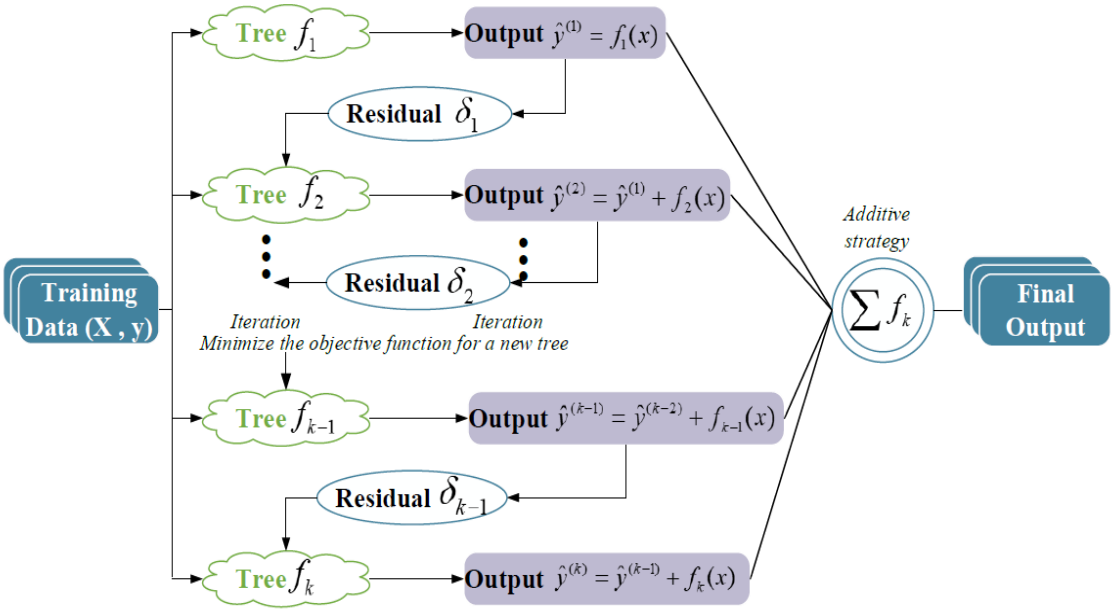

Figure 3.6 XGBoost Structure

The boosting depends on the additive training strategy, since each tree is built based on learning from the residual *d* of the previous tree, the prediction of *k*-th iteration can be expressed by the following equation:

$$\hat{y}_i^{(k)} = \hat{y}_i^{(k-1)} + f_k(x_i) \tag{3.8}$$

The model then is being optimized to decrease the prediction error, and the final output $\hat{y}_i$ is resulted by the weighted summation of trees (equation 3.9):

$$\hat{y}_i = \sum_{k=1}^{k} f_k(x_i)\,, \quad f_k \in \mathrm{F} \tag{3.9}$$

F is the space that includes all regression trees and K represents the number of trees. The objective function is presented by:

$$\zeta(\phi) = \sum_i l(y_i, \hat{y}_i) + \sum_i \Omega(f_k) \tag{3.10}$$

ϕ represents the learnable parameters in XGBoost, $l(y_i, \hat{y}_i)$ is the loss function which shows the error between predicted value $\hat{y}_i$ and the actual value y_i and $\Omega(f_k)$ is regularization used to disturb the model over fitting.

In many cases, the square loss function is used by XGBoost to measure error, and for quick objective optimization, XGBoost depends on the second derivative Taylor expansion of the loss function, the equation (3.11) presents the second derivative Taylor expansion after k-th iteration:

$$\Upsilon(\phi)^{(t)} \simeq \sum \left[l\left(y_i^{(t)}, \hat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i {f_t}^2(x_i) \right] + \Omega(f_t) \qquad (3.11)$$

where $g_i$and h_i represents the first & second derivative of the loss function. The loss function only depends on the first and second derivatives of each point of data. XGBoost uses variables in different areas of the output space benefitting from the additive tree boosting model. This model can perform effective feature selection and capture higher-order interactions.

(Morde, 2019) compared the performance of XGBoost with different algorithms such as: Random Forest, Logistic Regression, Gradient Boosting, the test was performed using 1 million data points with 20 features Figure 3.7.

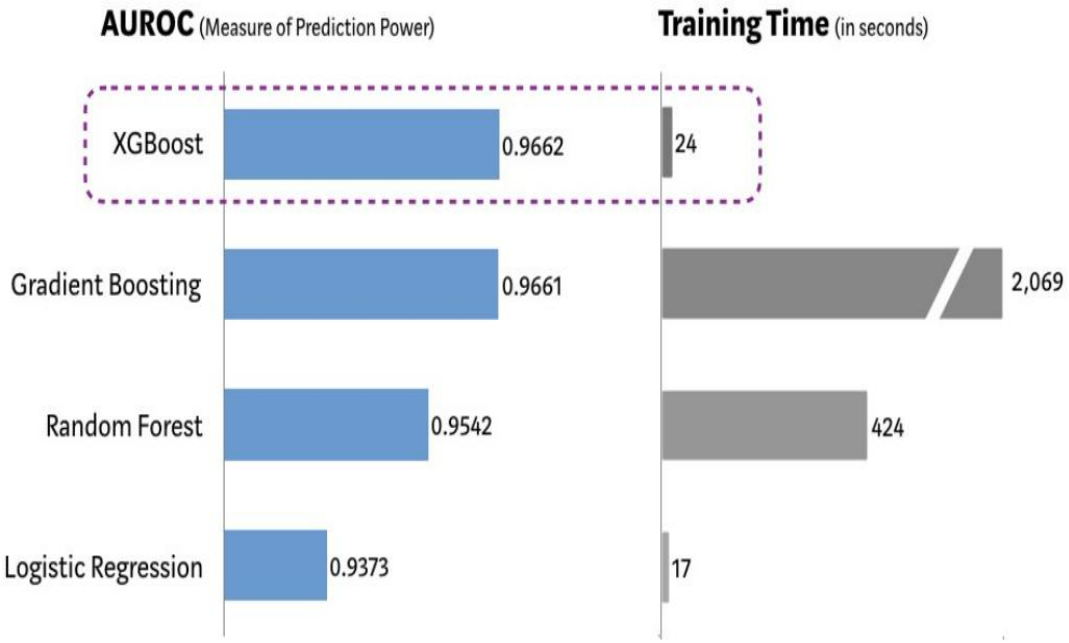

Figure 3.7 XGBoost vs. Other Machine Learning Algorithms

CHAPTER IV

EXPERIMENTAL WORK, RESULTS & DISCUSSION

The task of analyzing tweets' sentiments involved the following stages:

1. **Collect and preprocess the data**: as explained in the previous chapter, we used the Twitter API to collect a dataset of tweets on iPhone 13 and iPhone 14, then the collected tweets were preprocessed by removing unwanted characters, stemming and lemmatizing the text.
2. **Sentiment Analysis:** which means extracting a set of features from the tweet text that can be used by the sentiment classifier to make predictions. These features could include the presence of certain words or phrases, the use of certain part-of-speech tags, and the overall structure of the text.
3. **Training the Naive Bayes and XGBoost classifiers**: this was performed through:
    - Splitting the data into training and test sets.
    - Using the training data to train a Naive Bayes or XGBoost classifier to predict the sentiment of the tweets.
    - Evaluating the performance of the classifier on the test data.
4. **Applying the classifier to new data**: We used the trained classifier to predict the sentiment of new tweets collected using the Twitter API.
5. **Extracting the same set of features from the new tweets and feed them into the classifier to get a predicted sentiment label**.
6. **Analyzing and interpreting the results**: to calculate summary statistics and visualize the results of the sentiment analysis, and examine specific predictions made by the classifier and interpret the results in the context of the tweets being analyzed.

The first step was explained in details in the previous chapter, and in this one, the next steps was performed to get the results:

4.1 Sentiment Analysis:

Sentiment analysis is a natural language processing task that involves classifying the sentiment of a piece of text as positive, negative, or neutral. In the context of tweets, sentiment analysis can be used to analyze the tweets' sentiment of about a particular topic, event, or product in order to understand how people feel about it.

TextBlob is a Python library that provides a simple interface for performing sentiment analysis on text data. It uses a pre-trained machine learning classifier to predict the sentiment of a piece of text based on the presence and frequency of certain words and other features in the text.

The effectiveness of TextBlob for sentiment analysis of tweets depends on several factors, including the quality of the training data used to build the classifier, the complexity of the task, and the specific characteristics of the tweets being analyzed.

In general, sentiment analysis is a challenging task due to the complexity and variability of human language. It can be difficult for a machine learning model to accurately capture the nuances and subtleties of sentiment, especially in the context of social media where language is often informal and abbreviated. However, TextBlob and other NLP tools can still provide useful insights for sentiment analysis of tweets and other social media data. It can be especially useful for identifying broad trends and patterns in sentiment, even if the individual predictions are not always completely accurate.

The Sentiments analysis task includes the following steps:

- Text Vectorizations.
- Detect text polarity using TextBlob Python library.

4.1.1 Text Vectorization:

Machine learning classifiers only deal with numbers, which is why the results obtained from the previous stage should be transformed to a matrix of numbers, this is known as Text Vectorization. There are many ways to perform this step, such as: Bag of Words, (L1) Normalized Term Frequency, (L2) Normalized TF-IDF, Binary Term Frequency and Word2Vec.

In this study, Bag of Words (BoW) was selected, which is a technique that takes the whole corpus and assigns the vector representation to a given word. In BoW text words are referred to as tokens, so the process of representing words (or sentence) as BoW vector “a string of numbers” is called tokenization.

The words present are marked as one and absence as zero in the vector representation. The target feature is first encoded using the label encoder. The labels given are:

- Negative = 0
- Neutral = 1
- Positive = 2

4.1.2 Detecting Text Polarity:

The classifier used by TextBlob to determine the polarity of a piece of text is a supervised machine learning model. This means that it has been trained on a labeled dataset of text examples, with each example labeled as having a positive or negative sentiment. During training, the classifier learns to predict the sentiment label of a piece of text based on the presence and frequency of certain words and other features in the text. These features could include:

- The presence of certain positive or negative words (e.g. "love", "hate")
- The use of certain part-of-speech tags (e.g. adjectives)
- The overall structure of the text (e.g. the use of negation)

The classifier uses this training data to learn a set of rules or a mathematical model that can be used to predict the sentiment of a new piece of text based on its features.

When we call the sentiment method on a TextBlob object, the classifier uses the features of the text to make a prediction about its sentiment. It does this by applying the rules or model learned during training to the features of the text. The predicted sentiment is then returned as a polarity value between -1 and 1, as described above.

In this step, the tweets will be labeled with three main values: Positive, Negative and Neutral. We applied the sentiments analysis on both data sets, iPhone 13 and iPhone 14 and we got the following results:

Figure 4.1 presents the Tweets Polarities for iPhone 13 & iPhone 14:

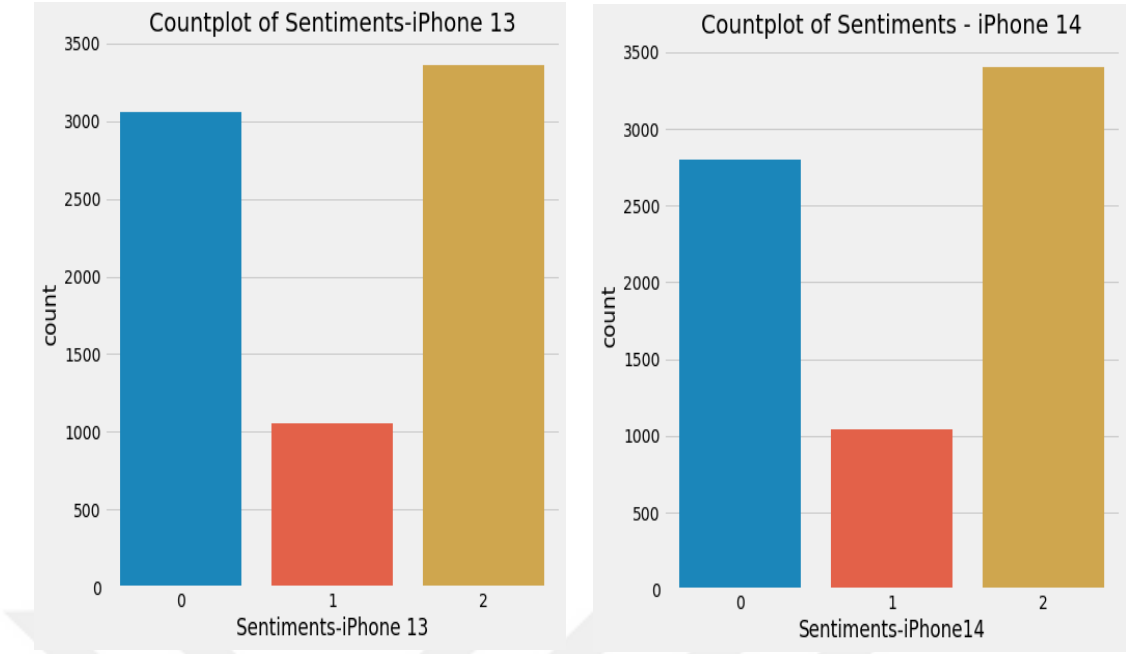

Figure 4.1 Tweets Polarities (with Counts)

Figure 4.2 shows the distribution of the three labels over the dataset:

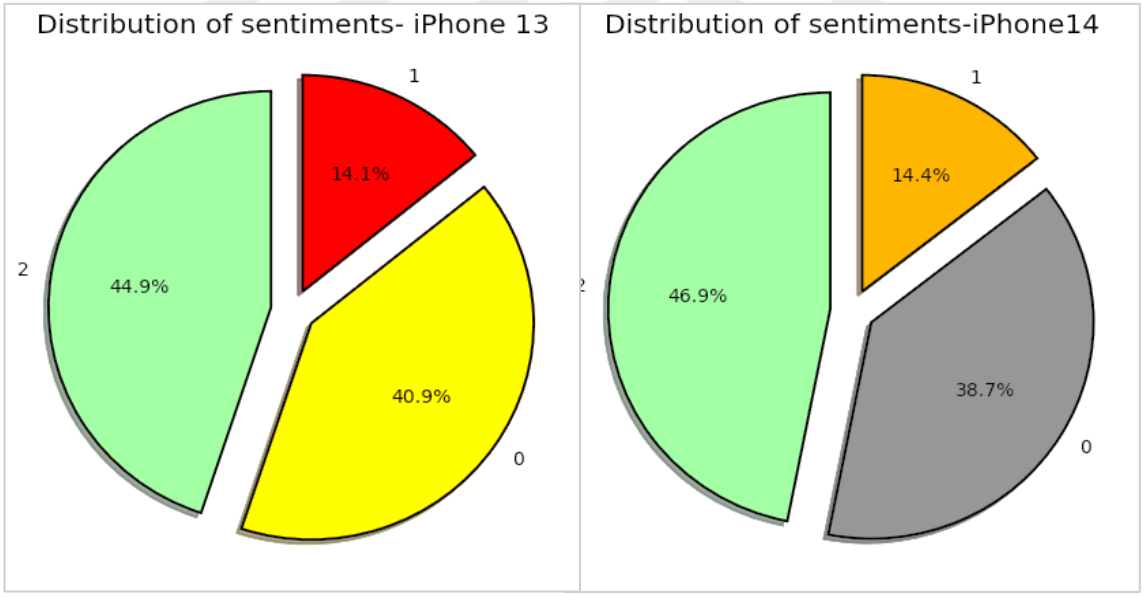

Figure 4.2 Tweets polarity distribution

4.2 Multinomial Naive Bayes Classifier Results:

To evaluate the performance of the classification model, we used the confusion matrix, which is particularly useful for comparing the predicted labels of a model with the true

labels of the test data. It can help us to understand how well the classifier is able to predict the sentiment of the tweets, and where it is making mistakes.

The confusion matrix typically consists of four quadrants that represent the following quantities:

- **True Positives (TP):** The number of tweets that the classifier correctly predicted as having a positive sentiment.
- **False Positives (FP):** The number of tweets that the classifier incorrectly predicted as having a positive sentiment.
- **True Neutrals (TNE):** The number of tweets that the classifier correctly predicted as having a neutral sentiment.
- **False Neutrals (FNE):** The number of tweets that the classifier incorrectly predicted as having a neutral sentiment.
- **False Negatives (FN):** The number of tweets that the classifier incorrectly predicted as having a negative sentiment.
- **True Negatives (TN):** The number of tweets that the classifier correctly predicted as having a negative sentiment.

From these quantities, a number of evaluation metrics can be calculated, including accuracy, precision, recall, and F1 score. These metrics can help to understand the overall performance of the classifier and identify areas for improvement.

The True classifications are distributed over the **left diagonal of the matrix,** while all other cells represent false classifications.

Figure 4.3 shows the confusion matrix for Naïve Bayes classifier for both iPhone 13 and iPhone 14 data sets.

TN		
	TNE	
		TP

The classification accuracy can be calculated directly from the confusion matrix by dividing the correct classification over the total of the matrix.

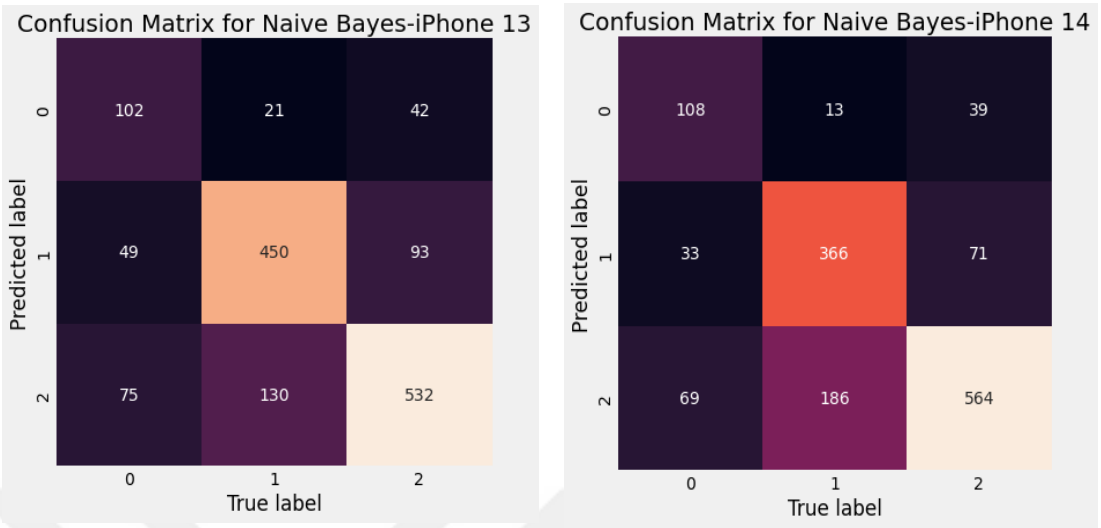

Figure 4.3 Confusion Matrix for Naive Bayes

- Accuracy of classification for iPhone 13 dataset = (102+450+532)/1494 =1084/1494 = 72.56%
- Accuracy of classification for iPhone 14 dataset= (108+366+564)/1449 =1038/1449 = 71.64%

4.3 XGBoost Classifier Results:

As mentioned in section 3.6, XGBoost is a highly effective machine learning method which gives higher accuracy compared with other classifiers, and this is the case in our study, as we can see from Figure 4.4 below.

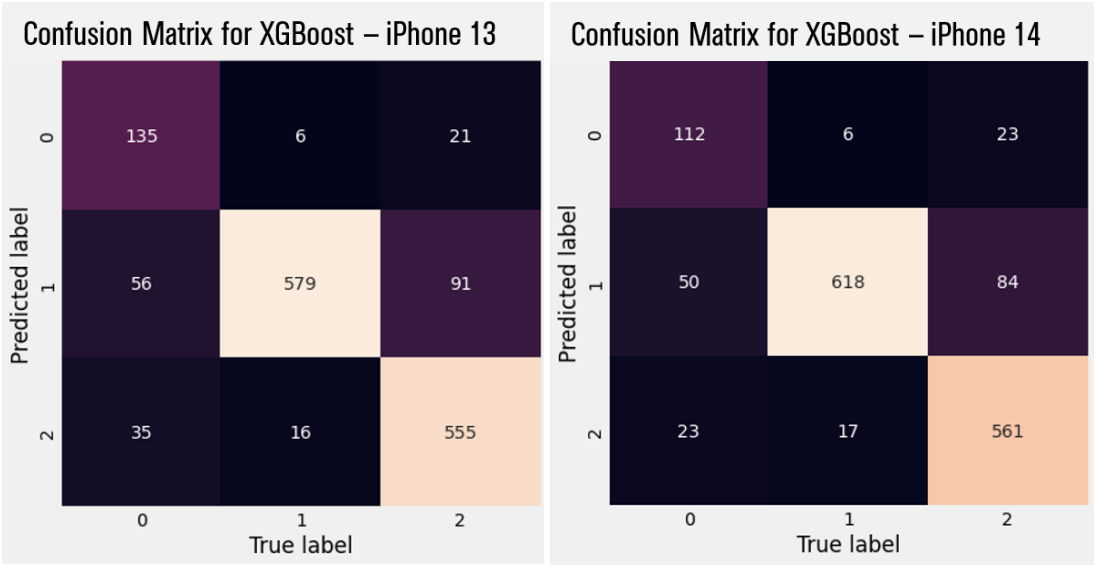

Figure 4.4 Confusion Matrix for XGBoost

- The accuracy of classification for iPhone 13 dataset = (135+579+555)/1494 =1269/1494 = **84.94%**
- Accuracy of classification for iPhone 14 dataset= (112+618+561)/1449 =1291/1449 = **89.09%**

4.4 Concept Drift Detection:

This task aims to find out whether the statistical models built on top of twitter data suffer from the problem of concept drift. The output of both models (Naïve Bayes and XGBoost) classifiers is the predicted value (the concept) of tweets sentiment, the model will receive the new streams of tweets and classify it according to sentiments categories defined already (positive, negative, or neutral).

The concept drift detection method depends on the prediction error rate, which was implemented through 4,041 tweets, which were fetched from 24/09/2021 to 11/12/2022, (Figure 4.5). This data to check for concept drift in both models – Naive Bayes & XGBoost.

	Date	tweet	sentiments
0	2020-01-07	iphone13 fact . may come headphone , charger &...	1
1	2020-01-07	news latest iphone , iphone 12 without charger...	2
2	2020-02-06	iphone 13 supposedly going new camera layout a...	2
3	2020-02-08	got ta skip next iphone cause ' fuxk 12 iphone13	0
4	2020-02-12	got new information regarding iphone 13. iphon...	2
...	...	...	...
7465	2021-09-24	batterylife course ! go longer huge leap batte...	2
7466	2021-09-24	else suppose get iphone 13 today , n't ? apple...	0
7467	2021-09-24	would love see visualization @ ups traffic pre...	2
7468	2021-09-24	thinking starting fundraiser buy iphone 13 iph...	0
7469	2021-09-24	new iphone13 battery amazing !	2

7470 rows × 3 columns

Figure 4.5 Data Set of Concept Drift Detection

A function was defined which splits the data into fragments and check for the accuracy, here, the step size is used as 50, which means the accuracy is checked for every 50

samples and the results are stored and further the results are plotted which helps determine the type of drift the model is facing.

The results obtained for both iPhone 13 and iPhone 14 are as follows:

1. The accuracy plot of Naive Bayes Classifier over time is shown in **Figure 4.6 (a):**

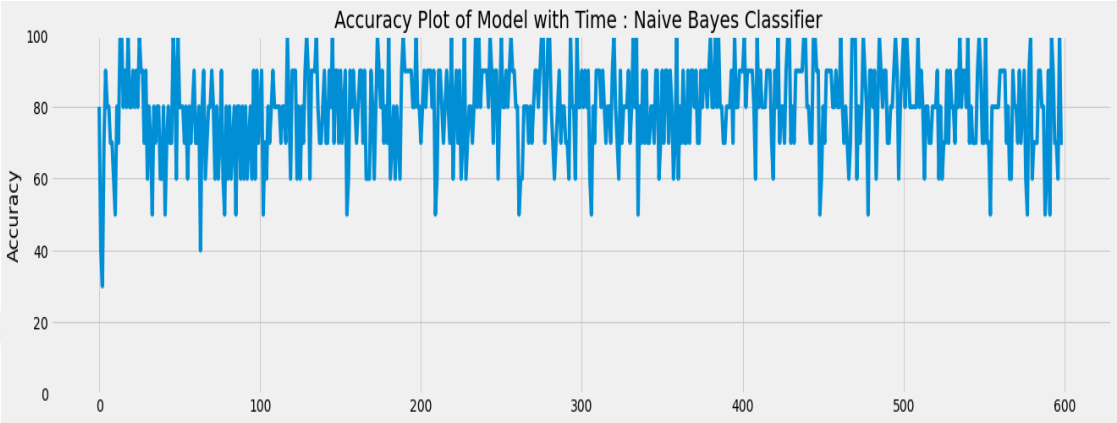

Figure 4.6 (a) Accuracy plot over time - Naive Bayes Classifier

2. The accuracy plot of XGBoost Classifier over time is shown in **Figure 4.6 (b):**

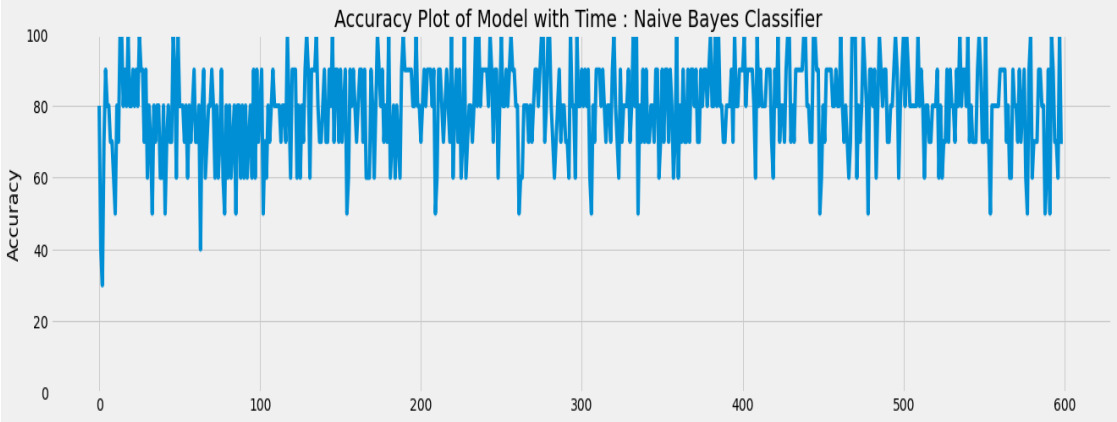

Figure 4.6 (b) - Accuracy plot over time - XGBoost Classifier

Figure 4.6 (a) and (b) illustrates the accuracy plot of both models with time, it is noticed that models are facing recurring drift over time since the accuracy is fluctuating between 40-100%.

4.5 Results Discussion

The study focused on understanding concept drift in technology through tweets mining, this was done depending two tweets data sets:

1. **iPhone 13 data set:** Almost 12K English Tweets, these tweets were cleaned, preprocessed and then split into two segments:
    - **i.** The first segment, about 7.5K Tweets, between 7/1/2020 and 24/09/2021 used to train and test both models, Naïve Bayes and XGBoost.
    - **ii.** The second segment, about 4K Tweets between 24/09/2021 and 11/12/2022 used for concept drift detection.
2. **iPhone 14 data set:** Almost 10K English Tweets, these tweets were cleaned, preprocessed and then split into two segments:
    - **i.** The first segment, about 8K Tweets, between 10/09/2021 and 24/02/2022 used to train and test both models, Naïve Bayes and XGBoost.
    - **ii.** The second segment, about 2K Tweets between 24/02/2022 and 11/12/2022 used for concept drift detection.

As presented in the results chapter, XGBoost classifier was giving a higher accuracy and better performance for both data sets, with **84.94%** and **89.09%** for iPhone 13 and iPhone 14 respectively, compared with: **72.56%** and **71.64%** for Naïve Bayes, this is due to the following reasons:

1. **First**, XGBoost is a tree-based ensemble learning algorithm that can handle high dimensional and sparse data more effectively than some other algorithms, such as Naive Bayes. This makes it well-suited for NLP tasks, where the data is often high dimensional and sparse due to the large number of unique words that can appear in the text.
2. **Second,** XGBoost can learn more complex decision boundaries compared to Naive Bayes, which is a relatively simple linear model. This means that it may be able to better capture the nuances and subtleties of sentiment in the tweets.
3. **Finally,** XGBoost allows for more fine-grained control over the model's complexity and behavior through the use of hyperparameters, which can help to improve its performance on the task at hand.

These results are in-line with many recent studies that reported higher performance and accuracy of XGBoost compared with other algorithms, as shown in Figure 3.6.

The sentiments analysis of iPhone 13 tweets shows that 44.9% of the tweets were positive, 40.9% are negative, while the remaining 14.1% are neutral. Similar results

were presented for iPhone 14 tweets, 46.9% Positive, 38.7% negative and 14.4% neutral.

The majority of tweets about the iPhone 13 and iPhone 14 are positive, it could suggest that people generally have a favorable view of the phones. This could be due to a variety of factors, such as the phone's design, performance, or new features. More positivity about iPhone 14 may be related to the developments that the company made on the new release. However, It's important to keep in mind that the results of sentiment analysis should be interpreted with caution, as they may not capture the full context or nuances of each individual tweet.

Overall, it's important to consider other factors in addition to sentiment analysis when trying to understand people's opinions about the iPhone 13 & 14. This could include looking at the content of the tweets themselves, as well as any other information or data that might be relevant to understanding people's views of the phone.

There are many different types of information that could be relevant to understanding people's opinions about the iPhone 13 & iPhone 14. Some examples could include:

- **Demographic information:** It could be useful to know the age, gender, location, or other characteristics of the people who are tweeting about the iPhones. This could give you a sense of who is most interested in the phone and how their views might differ.
- **Detailed content analysis:** Analyzing the content of the tweets in more detail could give a better understanding of what people like or dislike about the phone. For example, we could look for specific words or phrases that are used frequently in positive or negative tweets to get a sense of what features or aspects of the phone are most important to people.
- **External sources of information**: In addition to analyzing tweets, we could also look at other sources of information such as reviews, news articles, or forums to get a more comprehensive understanding of people's opinions about the phones.

Overall, it's important to consider a range of different sources of information when trying to understand people's opinions about the iPhones, as no single source will provide a complete picture.

For concept drift detection, the obtained results can be discussed as follows:

- The accuracy of models was fluctuating between 40% - 100%, which indicated the occurrence of **recurring drift**, which could suggest that the concept drift in the technology area (i.e., iPhone 13 and iPhone 14) was more incremental in nature, rather than radical. We may see more stable error values over time and less drastic changes in sentiment, since, as noticed from the accuracy plot, no dramatic changes in error values and sentiments.
- We have chosen the start of iPhone 14 tweets at the beginning of September 2021, which is the month that iPhone 14 was released in, and it was proofed that the sentiments of iPhone 14 were more positive which can support the theory that the incremental drift in iPhone 13 may led to new features, developments in iPhone 14, which means that the tweets reflected the changes in technology (the introduction of new features or improvements in the iPhone 14), this can be used to **understand technological change**.
- The accuracy fluctuation was also found in iPhone 14 accuracy plot, this can be understood since, although most of the sentiments were positive about the new phone, still there is a considerable share of negative tweets, which reflects that there are many users may not happy with the new phone, for several reasons.
- The results didn't show any dramatical change in accuracy or sentiments related to iPhone 13 in the period of iPhone 14 release, since there are many factors that can influence people's decisions to switch from one phone to another, and sentiment alone may not be a reliable predictor of phone adoption. For example, people may switch to a new phone for reasons unrelated to sentiment, such as a desire to upgrade to a newer model with better features or performance, or a change in carrier or service plan, and also, iPhone 14 is not a product that will completely replace iPhone 13, which means that not all users will try both phones.

  Overall, it's important to consider a range of different factors when trying to understand trends in phone adoption, and to rely on data rather than sentiment analysis alone.

From the results also we can conclude that using tweets as a source of data can be a useful way to gather information on a variety of topics, including the relationship

between concept drift and new technologies. However, it is important to carefully consider the **limitations** of using tweets as a source of data, as tweets may not be representative of the broader population and may be subject to biases or other confounding factors.

Here are a few potential limitations of using tweets as a source of data:

A. **Sample bias**: The individuals who tweet about a particular topic may not be representative of the broader population, and may have different attitudes, opinions, or behaviors than non-tweeters.
B. **Self-selection bias**: The individuals who choose to tweet about a particular topic may be more motivated or interested in the topic than those who do not tweet, which may skew the results.
C. **Limited context**: Tweets may provide limited context or background information, and may not provide a full picture of the attitudes or behaviors of the individuals who tweet.
D. **Incomplete or unbalanced data**: The dataset of tweets may be incomplete or unbalanced, and may not provide a full or accurate representation of the topic being studied.

Overall, it is important to carefully consider the limitations of using tweets as a source of data and to use a variety of approaches and methods in order to accurately understand the relationship between concept drift and new technologies.

CHAPTER V

CONCLUSION & FUTURE WORK

5.1 Conclusion:

While there are many studies in literature that focused on sentiment analysis, social media mining, concept drift in machine learning, and text mining but however, previous research has not adequately focused on understanding the innovational change caused by concept drift while performing sentiment analysis. A similar issue was raised by (Ozgun, Broekel, 2021) indicating that, so far, previous research has not adequately addressed the potential variations in news media's content and sentiment with respect to technologies.

Concept Drift is a phenomenon associated with classification models, especially in dynamic environments, which makes the model's prediction not sufficient anymore. Many studies in literature have focused on drift detection methods, systems, and corrective approaches, but limited studies focused on analyzing and/or understanding the change caused by concept drift over time, especially in the field of new inventions or modern technologies. Most of the studies refers to concept drift as negative effect, since the evaluation is from model accuracy perspective, but the change resulted by this phenomenon in technology field is not necessary to be negative, since it may lead to a new products or new inventions, and this was the main argument of this study.

In this study, we explored the use of sentiment analysis and concept drift detection of tweets to understand technological change in the form of concept drift in a technology area. We focused on two technology products, iPhone 13 and iPhone 14, and analyzed tweets related to these products using Naive Bayes and XGBoost algorithms for sentiment analysis, and error values for concept drift detection.

Our results showed that there was a recurring drift in the data, suggesting that the concept drift in the technology area was more incremental in nature, rather than radical. We also found that tweets can be used to understand technological change to some extent, as the sentiment analysis and concept drift detection reflected the changes in the technology products.

However, it is important to note that our study had several limitations. For example, the potential for bias or noise in the tweet data could have affected our results. Additionally, we only analyzed a limited number of tweets, which may not be representative of the broader population of tweets related to these technology products. Similarly, we only used two algorithms for sentiment analysis, which may not capture the full range of sentiment in the data. Finally, while our analysis provided some insight into the relationship between concept drift and technological change, more data and analysis would be needed to more definitively establish a causal relationship.

Despite these limitations, our study contributes to the understanding of concept drift and technological change by demonstrating the potential of sentiment analysis and concept drift detection of tweets as tools for exploring these phenomena. Future research could build on our findings by using larger and more diverse datasets of tweets, exploring different algorithms and methods, and investigating other sources of data to further understand concept drift and technological change.

5.2 Future Work:

This study presented an approach to understand the change in technology (innovation) based on tweets. The study focused on 270K tweets of ten technology companies, the focus was on analyzing the sentiment of these tweets and detecting/identifying the concept drift that actually occurred, and studying the results of this drift / change in an attempt to predict the pattern of results that could cause the deviation of the concept, which can in turn give some insights about future technological change or innovation.

The future work may include one or more of the following:

1. **Use a larger and more diverse dataset of tweets:** Expanding the dataset of tweets you analyze could provide a more comprehensive and robust analysis of concept drift and technological change. This could include tweets from a wider range of sources, such as different countries or languages, or tweets over a longer period of time.
2. **Explore different algorithms and methods for sentiment analysis and concept drift detection:** There are many different algorithms and methods available for sentiment analysis and concept drift detection. Testing and comparing different approaches could provide a more nuanced understanding of concept drift and technological change.
3. **Investigate other sources of data:** In addition to tweets, there are many other sources of data that could be used to study concept drift and technological change. For example, you could consider using patent data, news articles, or other types of social media data (e.g., Facebook, Reddit).
4. **Consider other factors that may influence concept drift and technological change:** There are many factors that may influence concept drift and technological change, such as market forces, regulatory environments, and technological infrastructure. Future research could explore the role of these and other factors in shaping concept drift and technological change.
5. **Develop new methods for detecting and analyzing concept drift:** There is always room for innovation and improvement in the methods used to detect and analyze concept drift. Future research could focus on developing new and more effective approaches for understanding concept drift and its relationship to technological change.

APPENDIX 1

PYTHON CODE

Loading Libraries

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib import style
style.use('fivethirtyeight')
from textblob import TextBlob
import re
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.tokenize import word_tokenize
from nltk.stem import PorterStemmer
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
from wordcloud import WordCloud
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report, confusion_
```

```
# Option Settings

pd.options.display.max_columns = None
pd.options.display.max_rows =100

import warnings
warnings.filterwarnings('ignore')
```

Loading Data

```
# Loading Data

df = pd.read_excel('iphone13.xlsx')
```

Text Pre-Processing

```
# Text Preprocessing

def data_processing(text):
    re.sub('[0-9]+', '', text)
    text = text.lower()
    text =re.sub(r'\@w+|\#','',text)
    text_tokens =word_tokenize(text)
    filtered_text = [w for w in text_tokens if not w in stop_words]
    return " ".join(filtered_text)
```

Stemming / Normalization

```
Using Porter Stemmer here

stemmer = PorterStemmer()

def stemming(data):
    text = [stemmer.stem(word) for word in data]
    return data
```

Calculating Polarity to Get Sentiments

```
# Using TextBlob for getting polarity

def polarity(text):
    return TextBlob(text).sentiment.polarity
```

```
# Defining function which assigns the sentiments based on polarity

def sentiment(label):
    if label < 0:
        return "Negative"
    elif label == 0:
        return "Neutral"
    elif label > 0:
        return "Positive"
```

Application of Operations

```
# Applying steps to create a final DataFrame for the modeling

def operations(df):
    df['tweet'] = df['Tweet'].apply(data_processing)
    df['tweet'] = df['tweet'].apply(stemming)
    df['polarity'] = df['tweet'].apply(polarity)
    df['sentiments'] = df['polarity'].apply(sentiment)

    df.drop_duplicates(inplace=True)
```

```
    data = df[['Date','tweet', 'sentiments']]
    data['sentiments'] = df['sentiments'].map({'Neutral'  : 0,
                                               'Negative' : 1,
                                               'Positive' : 2})

    data.sort_values(by = ['Date'], inplace=True)
    data.reset_index(inplace=True)

    data.drop('index', 1, inplace=True)

    return data
```

```
# Get the train data for new model

iphone13 = operations(train_iphone13)
```

```
iphone13
```

```
# Countplot of sentiments

fig = plt.figure(figsize=(7,7))
sns.countplot(x='sentiments', data = iphone13)

plt.xlabel("Count of Tweets")
plt.xlabel("Sentiments-iPhone 13")
plt.title("Countplot of Sentiments-iPhone 13")
```

```
# Pie chart of distribution

fig = plt.figure(figsize=(7,7))
colors = ("lightGreen", "gold", "red")
wp = {'linewidth':2, 'edgecolor':"black"}
tags = iphone13['sentiments'].value_counts()
explode = (0.1,0.1,0.1)
tags.plot(kind='pie', autopct='%1.1f%%', shadow=True, colors = colors,
          startangle=90, wedgeprops = wp, explode = explode, label = '')
plt.title('Distribution of sentiments- iPhone 13')
```

Models

```
# Preparing the Data for Model

from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.model_selection import train_test_split

bow_vectorizer = CountVectorizer(max_df = 0.90 ,min_df = 2, max_features = 1000, stop_words = 'english')

def model_prep(df):
    # Bag of Words
    bow = bow_vectorizer.fit_transform(df['tweet'])

    X_train, X_test, y_train, y_test = train_test_split(bow, df['sentiments'],
                                                        test_size = 0.2, random_state = 69)

    return X_train, X_test, y_train, y_test
```

```
# Assigning test train values

X_train, X_test, y_train, y_test = model_prep(train_iphone13)
```

```
# Check shape

X_train.shape, X_test.shape, y_train.shape, y_test.shape
```

Naive Bayes Model

```
from sklearn.naive_bayes import MultinomialNB  # Naive Bayes Classifier

model_naive = MultinomialNB()
model_naive.fit(X_train, y_train)
predicted_naive = model_naive.predict(X_test)
```

```
from sklearn.metrics import accuracy_score

score_naive = accuracy_score(predicted_naive, y_test)
print(f"Accuracy with Naive Bayes: {score_naive * 100 : .2f} %")
```

```
Accuracy with Naive Bayes:  72.56 %
```

```
from sklearn.metrics import confusion_matrix

plt.figure(figsize=(6,6))
mat_naive = confusion_matrix(y_test, predicted_naive)
sns.heatmap(mat_naive.T, annot=True, fmt='d', cbar=False)

plt.title('Confusion Matrix for Naive Bayes - iPhone 13')
plt.xlabel('True label')
plt.ylabel('Predicted label')
plt.savefig("confusion_matrix.png")
plt.show()
```

Concept Drift

```
# Applying the text preprocessing operations on test DataFrame

test_data_iphone13 = operations(test_iphone13)
```

```
test_data_iphone13
```

```
# Preparing test data for concept drift evaluation

from scipy.sparse import vstack, hstack

bow_test = bow_vectorizer.fit_transform(test_data_iphone13['tweet'])
bow_train = bow_vectorizer.fit_transform(train_iphone13['tweet'])

X_concat = vstack((bow_train, bow_test))
y_concat = np.concatenate((train_iphone13['sentiments'], test_data_iphone13['sentiments']), axis = 0)
```

```
X_concat.shape
```

```
# Calculating step by step accuracies

def acc_calculator(df, step, model):

    accuracies = []

    for i in range(0, 5000, step):
        y_pred = model.predict(X_concat[i:i+step])
        accuracy = accuracy_score(y_pred, y_concat[i:i+step]) * 100
        accuracies.append(accuracy)

    return accuracies
```

```
acc_naive = acc_calculator(X_concat, 10, model_naive)
```

```
# Plotting Concept Drift for Naive Bayes Classifier

plt.figure(figsize=(20,8))
plt.plot(acc_naive)
plt.title("Accuracy Plot of Model with Time : Naive Bayes Classifier ")
plt.ylim(0, 120)
plt.ylabel("Accuracy")
```

XGBoost

```
# XGBoost Classifier

from xgboost import XGBClassifier

model_xgb = XGBClassifier(max_depth = 6, n_estimators = 1000)
model_xgb.fit(X_train, y_train)
predicted_xgb = model_xgb.predict(X_test)

score_xgb = accuracy_score(predicted_xgb, y_test)
print(f"Accuracy with XGBoost Classifier: {score_xgb * 100 : .2f} %")
```

```
plt.figure(figsize=(6,6))
mat_xgb = confusion_matrix(y_test, predicted_xgb)
sns.heatmap(mat_xgb.T, annot=True, fmt='d', cbar=False)

plt.title('Confusion Matrix for XGBoost')
plt.xlabel('True label')
plt.ylabel('Predicted label')
plt.savefig("confusion_matrix.png")
plt.show()
```

```
# Accuracies for XGBoost

acc_xgb = acc_calculator(X_concat, 10, model_xgb)
```

```
# Plotting Concept Drift for Naive Bayes Classifier

plt.figure(figsize=(20,5))
plt.plot(acc_xgb)
plt.title("Accuracy Plot of Model with Time : XGBoost Classifier ")
plt.ylim(0, 100)
plt.ylabel("Accuracy")
```

Same code was applied on iPhone 14 dataset.

REFERENCES

[1] Abbasi, A., Chen, H., Salem, A. (2008). Sentiment Analysis in Multiple Languages: Feature Selection for Opinion Classification in Web Forums. *ACM Transactions on Information Systems*, 1-34.

[2] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., Passonneau, R. (2011). Sentiment analysis of twitter data. *Proceedings of the workshop on languages in social media. Portland, Oregon*: Association for Computational Linguistics.

[3] Aggarwal, C. C. (2007). *Data Streams: Models and Algorithms* . New York, NY: Springer .

[4] Agrahari, S., Singh, A. K. (2021). Concept Drift Detection in Data Stream Mining: A literature review. J*ournal of King Saud University – Computer and Information Sciences*.

[5] Alencia, C., Nizar, A., Ayuning, N., Herkules. (2018). Sentiment Analysis of Online Auction Service Quality on Twitter Data: A case of E-Bay. *The 6th International Conference on Cyber and IT Service Management (CITSM 2018). Medan, Indonesia*.

[6] Baena-Garcıa, M., Campo-Avila, J. d., R. Fidalgo, A. B., Gavalda, R., Morales-Bueno, R. (2006). Early drift detection method. *4th Int. Workshop Knowledge Discovery from Data Streams*.

[7] Barzenji, H. S. (2021). Sentiment Analysis of Twitter Texts Using Machine Learning Algorithms. *Academic Platform Journal of Engineering and Science*, **9(3)**, 461-471.

[8] Bayram, F., Ahmed, B. S., Kassler, A. (2022). From concept drift to model degradation: An overview on performance-aware drift detectors. *Knowledge-Based Systems* .

[9] Bhuvaneswari, M., Srividhya, V. (2017). Enhancing the Sentiment Classification Accuracy of Twitter Data using Machine Learning Algorithms. *CORE UK*.

[10] Bifet, A., Gavaldà, R. (2007). Learning from Time-Changing Data with Adaptive Windowing. *The 2007 SIAM International Conference on Data Mining (SDM)*, (443 - 448).

[11] Cheng, F., Yang, C., Zhou, C., Lan, L., Zhu, H., Li, Y. (2020). Simultaneous Determination of Metal Ions in Zinc Sulfate Solution Using UV–Vis Spectrometry and SPSE-XGBoost Method. *Journal of Sensors* **20 (17)**.

[12] Dave, K., Lawrence, S., Pennock, D. M. (2003). *Mining the Peanut Gallery-Opinion Extraction and Semantic Classification of Product Reviews*.

[13] *Proceedings of the 12th International Conference on World Wide Web, WWW 2003. Budapest, Hungary*.

[14] Devi, B., Bai, V., Ramasub, S. (2020). Sentiment Analysis on Movie Reviews. In *Emerging Research in Data Engineering Systems and Computer Communications* (pp. 321-328). India: Springer Nature Singapore Pte Ltd.

[15] Ditzler, G., Polikar, R. (2013). ncremental Learning of Concept Drift from Streaming Imbalanced Data . *IEEE Transactions on Knowledge and Data Engineering*.

[16] Elwel, R., Polikar, R. (2011). Incremental Learning of Concept Drift in Nonstationary Environments. *IEEE Transactions on Neural Networks,* **22(10)**, 1517-1531.

[17] Fredstrom, A., Parida, V., Wincent, J., Sjodin, D., Oghazi, P. (2022). What is the Market Value of Artificial Intelligence and Machine Learning? The Role of Innovativeness and Collaboration for Performance. *Technological Forecasting & Social Change,* **180 (C)-** 121716**.**

[18] Gama, J., Medas, P., Castillo, G., =Rodrigues, P. (2004). Learning with Drift Detection. In *Advances in Artificial Intelligence – SBIA 2004* (pp. 286–295). Berlin, Heidelberg: Springer.

[19] Gama, J., Zliobaite, I., Bifet, A., Pechenizkiy, M., Bouchachia, A. (2014). A Survey on Concept Drift Adaptation. *ACM Computing Surveys*, **1(1)**, 35.

[20] Gangawane, A., Torvi, H. B. (2017). *Opinion Mining and Sentiment Analysis on Twitter*. **6(7).**

[21] Gemaq, R., Costa, A. F., Giusti, R., Santos, E. (2020). An overview of unsupervised drift detection methods. *WIREs Data Mining And Knowledge Discovery* by Wiley Periodicals LLC, **10(6)**.

[22] Gözüaçık, Ö., Can, F. (2021). Concept learning using one-class classifiers for implicit drift detection in evolving data streams. *Artificial Intelligence Review*, 3725–3747.

[23] Gu, F., Zhang, G., Lu, J., Lin, C.-T. (2019). Concept drift detection based on equal density estimation. *International Joint Conference on Neural Networks (IJCNN)*. Vancouver, BC.

[24] Gupta, A., Pruthi, J., Sahu, N. (2017). Sentiment Analysis of Tweets using Machine Learning Approach. *International Journal of Computer Science and Mobile Computing*, **6(4)**, 444 – 458.

[25] Han, J., Kamber, M., Pei, J. (2012). Data Mining- *Concepts and Techniques.* Elsevier Inc.

[26] Hu, M., Liu, B. (2004). Mining and summarizing customer reviews. *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, (p. 168).

[27] Janardan, S. M. (2017). Concept drift in Streaming Data Classification: Algorithms, Platforms and Issues. *5th International Conference on Information Technology and Quantitative Management (ITQM2017).* India: Faculty of Mathematics and Informatics, Vilnius University,.

[28] Kantardzic, M. (2020). *DATA MINING Concepts, Mode, Methods, and Algorithms (Vol. THIRD EDITION).* New Jersey: John Wiley and Sons, Inc., Hoboken.

[29] Kononenko, I., Kukar, M. (2007). *Machine Learning and Data Mining. In M. K. Igor Kononenko, Machine Learning and Data Mining*. Woodhead Publishing Limited.

[30] Learning in the presence of concept drift and hidden contexts. (1996). *Machine Learning* , **23(1)**, 69-101.

[31] Li, P. (2010). Robust LogitBoost and Adaptive Base Class (ABC) LogitBoost. *Twenty-Sixth Conference on Uncertainty in Artificial Intelligence (UAI2010).*

[32] Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., Zhang, G. (2019). Learning under Concept Drift: A Review. *IEEE Transactions on Knowledge and Data Engineering*, **31(12)**, 2346 - 2363.

[33] Lu, N., Guangquan Zhang, Lu, J. (2007). Concep tdrift detection via competence models. *Artificial Intelligence*, **11(24)**, .

[34] Maciel, B. I., Santos, S. G., Barros, R. S. (2015). A Lightweight Concept Drift Detection Ensemble. *IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*. Vietri sul Mare, Italy.

[35] Maimon, O., Rokach, L. (2010). *Data Mining and Knowledge Discovery Handbook (Vol. Second Edition)*. New York: Springer .

[36] Morde, V. (2019, April). Retrieved from Towards Data Science: https://towardsdatascience.com/https-medium-com-vishalmorde-xgboost-algorithm-long-she-may-rein-edd9f99be63d

[37] Ogul, H., Ekmekciler, E. (2012). Two-way collaborative filtering on semantically enhanced movie ratings. *Proceedings of the ITI 2012 34th International Conference on Information Technology Interfaces. Cavtat, Croatia*.

[38] Ozgun, B., Broekel, T. (2021). The geography of innovation and technology news - An empirical study of the German news media. *Technological Forecasting & Social Change*, **167(1)** -120692

[39] Qahtan, A. A., Alharbi, B., Wang, S., Zhang, X. (2015). A PCA-Based Change Detection Framework for Multidimensional Data Streams: Change Detection in Multidimensional Data Streams. *The 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (pp. 935–944).

[40] Rianto, Mutiara, A. B., Wibowo, E., Santosa, P. I. (2021). Improving the accuracy of text classifcation using stemming method, a case of non-formal Indonesian conversation. *Journal of Big Data*, **8(26)**.

[41] Saberi, B., Saad, S. (2017). Sentiment Analysis or Opinion Mining: A Review. I*nternational Journal on Advanced Science, Engineering and Information technology*. **7 (5)**. (2088-5334).

[42] Sagayam, R., S.Srinivasan, Roshni, S. (2012). A Survey of Text Mining: Retrieval, Extraction and Indexing Techniques. *International Journal Of Computational Engineering Research (ijceronline.com)* **2(5)**, 1443-1446.

[43] Schuh, G., Reinhart, G., Prote, J.-P., Sauermann, F., Horsthofer, J., Oppolzer, F., Knoll, D. (2019). *Data Mining Definitions and Applications for the Management of Production Complexity.* Procedia CIRP, 874-879.

[44] Tianqi Chen, C. G. (2016). XGBoost: A Scalable Tree Boosting System. *22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[45] Vasudevan, S. (2017). *Enhancing the Sentiment Classification Accuracy of Twitter Data using Machine Learning Algorithms*. *In Statistical Approaches in Multidisciplinary Research* (p. Chapter 21). India: SHANLAX PUBLISHER.

[46] Vijayalakshmi, M. (2015). Identifying Concept-Drift in Twitter Streams. *International Conference on Advanced Computing Technologies and Applications* (ICACTA-2015). Mumbai, India.

[47] Wang, H., Abraham, Z. (2015). Concept Drift Detection for Streaming Data. *International Joint Conference of Neural Networks 2015*.

[48] Wang, J.-H., Liu, T.-W. (2017). Improving sentiment rating of movie review comments for recommendation. 2017 *IEEE International Conference on Consumer Electronics - Taiwan (ICCE-TW). Taiwan* .

[49] Weiss, S. M., Indurkhya, N., Zhang, T., Damerau, F. (2005). *Text Mining: Predictive Methods for Analyzing Unstructured Information.* New York: Springer .

[50] Widmer, G., Kubat, M. (1996). *Learning in the Presence of Concept Drift and Hidden Contexts. Machine Learning*, 69-101.

[51] Yang, L., McClean, S., Donnelly, M., Burke, K., Khan, K. (2022). *Detecting and Responding to Concept Drift in Business Processes. Algorithms*, 1, 15, 174.

[52] Zhou, X., Tao, X., Yong, J., Yang, Z. (2013). Sentiment Analysis on Tweets for Social Events. *IEEE 17th International Conference on Computer Supported Cooperative Work in Design. Whistler, BC, Canada*.

CURRICULUM VITAE

Mohamad NACİ

Education:

2005 – 2010 Bachelor Degree in Electrical Engineering University of Aleppo

Work Experiences:

- **GOAL International, Gaziantep**

  Grants Manager **June 2022 till now**

  - Support & lead the development of concepts and proposals, in accordance with GOAL processes, which fit within GOAL Syria's strategic objectives. Ensure a comprehensive grant reporting process is in place at the outset of each new grant – including the creation of templates and sharing of requirements, roles, responsibilities, and timelines.
  - Manage donor reporting processes through the coordination and drafting of high-quality donor reports for the GOAL Syria grant portfolio
  - Ensure compliance with GOAL and donor requirements and timely submission to donors, ensure quality standards are met.
  - Lead the grant management meetings for the assigned grants. Contribute to regular project review processes as required.
  - Respond to ad-hoc requests from GOAL team, GOAL HQ, donors, and other stakeholders about GOAL Syria programs.

- **MIDMAR Organization, Gaziantep**

  Grants & Partnerships Manager **June 2020 till June 2022**

  - Identify new partners/donors and open communication channels for possible cooperation opportunities..
  - Review and submit monthly, quarterly, semi and final reports on time based on donor requirement.

- Lead the products development process (includes writing and/or reviewing concept notes, proposals & budgets), and supervise the external/internal communication within the design stage

➢ **UOSSM – Union des Organisations de Secours et Soins Médicaux Organization, Gaziantep**

Grants & Compliance Coordinator **May 2017 till June 2020**

- Review and submit monthly, quarterly, semi and final reports on time based on donor requirement.
- Ensure the project implementation according to the approved internal policies and donor's requirement.
- Lead the grants team who follow up with donors/partners during the project implementation.
- Supervise the OCAs with donors/partners and coordinate with relevant departments

COMPUTER SKILLS EXPERIENCE

➢ Experience in the following software & programming languages:

- Microsoft Office (All versions).
- Microsoft Outlook.
- Python.

CERTIFICATES

2022-04	**Program Management Professional (PgMD Pro)** *PM4NGOs, UK*
2020-07	**Project Management Professional (PMD Pro)** *APMG International, UK*
2021-09	**Training of Trainers (ToT)** *International Board of Certified Trainers*
2022-02	**Results-Based Management (Advanced Level)** *UN Women*