



REVA
UNIVERSITY

Bengaluru, India

A Project Report on
Sentiment Analysis on Credit Cards
using Online Reviews

Submitted in Partial Fulfilment for Award of Degree of
Master of Business Administration
In Business Analytics

Submitted By
Madhukeshwar. R K
R19MBA57

Under the Guidance of
Ratnakar Pandey
Leading ML and Analytics for Customer Services at Amazon

REVA Academy for Corporate Excellence - RACE
REVA University
Rukmini Knowledge Park, Kattigenahalli, Yelahanka, Bengaluru - 560 064
race.reva.edu.in

August, 2022



Candidate's Declaration

I, **Madhukeshwar R K** hereby declare that I have completed the project work towards the first year of **Master of Business Administration in Business Analytics** at, **REVA University** on the topic entitled **Sentiment Analysis on Credit Card using Online Reviews** under the supervision of **Ratnakar Pandey, Leading ML and Analytics for Services at Amazon**. This report embodies the original work done by me in partial fulfillment of the requirements for the award of a degree for the academic year **2022**.

Place: Bengaluru

Name of the Student: Madhukeshwar R K

Date: 27-August-2022

Signature of Student



Certificate

This is to Certify that the project work entitled **Sentiment Analysis on Credit Card Using Online Reviews** carried out by **Madhukeshwar R K** with **SRN R19MBA57**, is a bonafide student at REVA University, is submitting the second-year project report in fulfillment for the award of **Master of Business Administration** in Business Analytics during the academic year 2022. The Project report has been tested for plagiarism and has passed the plagiarism test with a similarity score of less than 15%. The project report has been approved as it satisfies the academic requirements in respect of the project work prescribed for the said degree.

Signature of the Guide

Name of the Guide

Mr. Ratnakar Pandey

Signature of the Director

Name of the Director

Dr. Shinu Abhi

External Viva

Names of the Examiners

1. Vaibhav Sahu, Strategic Cloud Engineer, Google
2. Abhishek Sinha, Data Science Manager, Capgemini

Place: Bengaluru

Date: 27-August-2022



Acknowledgment

I am highly indebted to **Dr. Shinu Abhi**, Director, and Corporate Training for their guidance and constant supervision as well as for providing necessary information regarding the project and for their support in completing the project.

I would like to thank my project guide **Mr. Ratnakar Pandey** and my team members **Mr. Ravikumar** and **Mr. Surendra** for the valuable guidance provided to understand the concept and execute this project. It is gratitude towards all other mentors for their valuable guidance and suggestion in learning various data science aspects and for their support. I am thankful for my classmates for their aspiring guidance, invaluable constructive criticism, and friendly advice during the project work.

I would like to acknowledge the support provided by Hon'ble Chancellor, **Dr. P Shyama Raju**, Vice Chancellor **Dr. M. Dhananjaya**, and Registrar **Dr. N. Ramesh**. It is sincere thanks to all members of the program office of RACE who are supportive of all requirements from the program office.

It is my sincere gratitude towards my parents, and my family for their kind co-operation and encouragement which helped me in the completion of this project.

Place: Bengaluru

Date: 27 August 2022



Similarity Index Report

This is to certify that this project report titled **Sentiment Analysis on Credit Card using Online Reviews** was scanned for similarity detection. Process and outcome are given below.

Software Used: Turnitin

Date of Report Generation: 29 July 2022

Similarity Index in %: 10%

Total word count: 5295

Name of the Guide: Ratnakar Pandey

Place: Bengaluru

Name of the Student: Madhukeshwar R K

Date: 27 August 2022

Signature of Student

Verified by: M N Dincy Dechamma

Signature

Dr. Shinu Abhi,

Director, Corporate Training

List of Abbreviations

Sl. No	Abbreviation	Long Form
1	NLP	Natural Language Processing
2	ML	Machine Learning
3	CRISP-DM	Cross Industry Standard Process for Data Mining
4	NLTK	Natural Language Toolkit
5	TF-IDF	Term Frequency Inverse Document Frequency
6	EDA	Exploratory Data Analysis
7	BOW	Bags of Words
8	AFFIN	Arup Finn Nielsen
9	VADER	Valence Aware Dictionary and Sentiment Reasoner
10	RNN	Recurrent Neural Networks
11	Word2 Vec	Word of Vector
12	BERT	Bidirectional Encoder Representation for Transformer
13	SWOT	Strengths Weaknesses Opportunities Threats
14	LUIS	Language Understanding Service
15	QNA	Question and Answer Pairs

List of Figures

No.	Name	Page No.
Figure No. 1.1	Credit Card Swipes over time	11
Figure No. 5.1	CRISP-DM Framework	18
Figure No. 7.1	NLP Flow Diagram	19
Figure No. 7.2	Reviews Text acquired from web	19
Figure No. 7.3	Parameters used to acquire tweets from Twitter	19
Figure No. 7.4	Pre-processing data extract from Web	20
Figure No. 7.5	Pre-processing data from Twitter	20
Figure No. 7.6	Data Pipeline	20
Figure No. 8.1	The number of words	21
Figure No. 8.2	Number of Characters	21

Figure No. 8.3	Number of stop words	21
Figure No. 8.4	The number of special characters	22
Figure No. 8.5	Number of numeric	22
Figure No. 8.6	Lower Case	22
Figure No. 8.7	Removing Punctuations	23
Figure No. 8.8	Removing Stopwords	23
Figure No. 8.9	Spelling Corrections	23
Figure No. 8.10	Tokenization	23
Figure No. 8.11	Stemming	24
Figure No. 8.12	Lemmatization	24
Figure No. 8.13	N-grams	24
Figure No. 8.14	TF IDF	24
Figure No. 8.15	Bags of Words	25
Figure No. 8.16	Most Common Words	25
Figure No. 8.17	Word Frequency	25
Figure No. 8.18	Word Cloud	26
Figure No. 8.19	Word Frequency Distribution	26
Figure No. 8.20	Pre-processed Data for Recommendation Engine	28
Figure No. 8.21	Dataset with reduced features for Recommendation Engine	28
Figure No. 8.22	Dataset with binned values of Polarity	28
Figure No. 8.23	Dataset for Recommendation Engine	28
Figure No. 8.24	Dataset .json file used for training Chat Bot	28
Figure No. 9.1	EDA of Lexicon Approach	29
Figure No. 9.2	Models used for Lexicon Approach	29
Figure No. 9.3	EDA of Text Blob Approach	30
Figure No. 9.4	Models for Text Blob Approach	30
Figure No. 9.5	EDA of VADER Approach	31
Figure No. 9.6	Models for VADER Approach	31
Figure No. 9.7	EDA of TF-IDF Approach	32
Figure No. 9.8	Models for TF-IDF Vectorizer	32
Figure No. 9.9	BERT Model	34
Figure No. 9.10	KNN Based Collaborative Model	34

Figure No. 9.11	Bot using NLTK and Keras	35
Figure No. 10.1	Comparison of Accuracies	36
Figure No. 10.2	Comparison of Accuracies of Text Blob Polarity Model	37
Figure No. 10.3	Decision Tree Confusion Matrix – Text Blob	37
Figure No. 10.4	KNN Collaborative Filtering Models Recommendations	38
Figure No. 10.5	Loss and Accuracy of the Retrieval Model for Chat Bot	38
Figure No. 11.1	Deployment Model for Tableau dashboard and Recommendation Engine	39
Figure No. 11.2	Deployment Model for Retrieval Chat Bot	39
Figure No. 12.1	SWOT Analysis	40
Figure No 12.2	Credit Card Sentiment Analysis Dashboard	40
Figure No 12.3	Credit Card Dashboard by Regions in India	41
Figure No. 12.4	Recommendations for general credit card	41
Figure No. 12.5	Credit card Chat Bot Responses	41
Figure No. 13.1	Bot Architecture on Azure	42

List of Tables

No.	Name	Page No.
Table No. 10.1	Accuracies of Modelling	36
Table No. 10.2	Decision Tree Classifier – Result	36

Abstract

Sentiment analysis is a process of computationally finding, classifying, and categorizing opinions expressed on the block of text, to determine whether his / her sentiment towards a particular topic, product, etc. is positive, negative, or neutral and on emotions happy, sad, angry, etc. It combines Machine Learning and Natural Language Processing to achieve this.

With the increase in usage of web and social media services, views and experiences on products or services shared by online users have increased. It acts as the main source for text analytics data on which sentiment analysis can be performed to gain insights by organizations to measure their ability on the impact of marketing strategies by finding the sentiments towards their products. This helps customers to take better decisions through proper analysis of the collected sentiments.

Credit cards are becoming less of a source of credit and more of a transactional platform. Many credit card users are switching from credit seekers to regular users of the credit card in place of cash. More users are using credit cards as a transactional medium due to convenience. Banks also offer a platform where one can track their spending and so adjust their spending accordingly. Based on the user's transactional history of credit cards, lenders offer many offers, promotions, and discounts on their next transactions.

The scope of this project is to perform text analysis and sentiment analysis on the reviews and tweets collected from websites and Twitter and to develop tableau dashboards, a simple chatbot, and a credit card recommendation engine for the users, to choose the right credit card for right offer or discounts offered by the credit card issuer.

Keywords: Text Mining, Sentiment Analysis, Natural Language Processing, Chat Bot, Recommendation engine.

Contents

Candidate's Declaration.....	2
Certificate.....	3
Acknowledgment	4
Similarity Index Report.....	5
List of Abbreviations	6
List of Figures	6
List of Tables	8
Abstract.....	9
Chapter 1: Introduction	11
Chapter 2: Literature Review	13
Chapter 3: Problem Statement	16
Chapter 4: Objectives of the Study	17
Chapter 5: Project Methodology	18
Chapter 6: Business Understanding	19
Chapter 7: Data Understanding.....	20
Chapter 8: Data Preparation.....	22
Chapter 9: Modeling	30
Chapter 10: Model Evaluation	37
Chapter 11: Deployment	40
Chapter 12: Analysis and Results	41
Chapter 13: Conclusions and Recommendations for future work	44
Bibliography	45
Appendix.....	47
Plagiarism Report.....	47
Publications in a Conference Presented	50
Publications in a Journal	51
Github Link.....	59

Chapter 1: Introduction

Credit cards have become such an everyday part of credit cardholder's life, although it is surprising how long the concept has been around. The history of credit cards starts in 1887, with the idea of using a card to make purchases described in the novel *Looking Backward*, written by Edward Bellamy. Credit cards were first used in the 1920s in the U.S as a successor to many other forms of merchant credit. Credit cards were first intended to sell gasoline to the growing number of drivers on the road. It was not until 1938 that many companies began accepting these cards from other businesses. By 1921, Western Union also started to issue charge cards to regular customers. During this time a lot of charge cards were simply printed on card stock, so counterfeiting was a big concern (*History of the Credit Card*, 2017).

Credit card holders can buy anything on credit within the limit and pay later, even on the high value of the credit. On high-value purchases, the concept of converting the total purchase amount into low-cost EMIs has revolutionized the card holder's experience of shopping. Traveling anywhere without carrying much money, is one of the most accepted methods of payment. Some credit cards at the airport or railway station give cardholders a unique experience through complimentary lounge access and priority check-in. Along with the discounts on food at the restaurants. Some cards also cover comprehensive travel insurance coverage. Few credit cards will be handy during a financial emergency for withdrawal of cash with no interest charged for up to 45 to 50 days. Discounts on credit cards get extended on movie tickets, online shopping, health, and wellness outlets, and waivers at petrol/diesel pump across the country.

Learning to use a credit card and a better understanding of the credit card period and its repay the amount within the period helps credit card holders to boost their CIBIL score and help them to get the eligible amount of loan without any difficulty.

“Credit card spending hit Rs 2 Lakh crore high in 2021. Spending on credit cards had taken a severe hit during the first 9 months of the pandemic between March 2020 to December 2020. Spending on debit cards outstripped credit for this period. October has begun with a bang with the reopening of malls and online sales” as per Figure No 1.1.

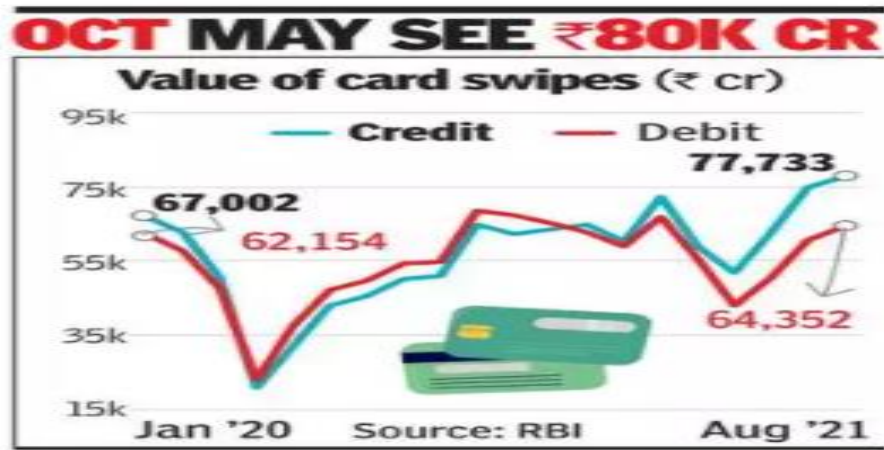


Figure No 1.1 Credit Card swipes over time. (Times Of India "*Credit Card Spends Hit Rs 2 Lakh Crore High in Q2*" - Times of India)

Chapter 2: Literature Review

This section examines the extant literature available on the need for developing a recommendation system based on sentiment analysis with natural language techniques in the credit card market.

Given the fierce competition in the global market the credit card industry is highly vulnerable to customer power in the emerging e-payment and fintech industry. Innovation in financial services to provide a personalized experience to customers with a better offer on credit card transactions become inevitable (Umuhoza et al., 2020). Mining credit card reviews can help the banks to find interesting patterns among different variables that may be used in the future to design better products (Zaza & Al-Emran, 2016).

The digital transformation in the banking sector has enforced e-payment customers to purchase products online by staying at home and due to the proliferation of online stores, online reviews have also increased as the source of information on product quality and durability. Opinion mining on these online reviews helps customers to make a better decision on the purchase (Mittal & Agrawal, 2022).

Online shopping has become increasingly popular due to the variety of products, lower prices, availability of different models/brands, and fast logistic systems. The explosion of offers has forced the shoppers to make use of debit/credit cards and to avail of this cashback and discounts offered by card issuers. Credit card holders typically aggregate reward points through various offers from multiple credit cards. Cashback or rewards acquired from credit card transactions varies in percentage from the issuer of the credit cards, identifying the best reward or cashback for a given card is difficult (Javkar et al., 2016).

Post shopping, shoppers provide their ratings, review, and emotions on websites which becomes the main source for purchaser's sentiments data generation. Multiple tools and techniques are available in the market for automatically classifying the sentiments for user-generated data. Sentiment analysis helps users to make better purchases through their collective analysis of sentiments. In deep learning models, the network learns to extract the features while

the learning/training process. Word2vec modeling technique uses CNN to get trained and to classify the sentiments on reviews collected (Shah, 2021).

User text data can also be fed to a stochastic learning algorithm that analyses and classifies the feedback as negative, positive, and neutral and provides recommendations to shoppers for their next purchases (P, 2020). Lexicon, machine learning, or a hybrid combination of both are the most commonly used approach (Ahmad et al., 2020).

Lexicon Based algorithms can classify the user sentiment through polarity score or using a machine learning classifier to identify specific text into a sentiment class. Two problems to be solved here are subjectivity classification; a text is subjective or objective and polarity classification; the text is positive or negative or neutral (*Sentiment Analysis in Banking - Maveric Systems*, n.d.). The lexicon-based approach is easy to understand and implement (Chakrabarti et al., 2018). However, user-shared reviews raise challenges due to insufficient coverage of emotions expressed. With an unsupervised approach, accuracy is determined by the classifier that might need modifications or negations (Asghar et al., 2017).

Recommendation systems typically are classified into content, collaborative, and hybrid-based recommendations. When properties of targets are considered for the recommendation it is called content based. When the system recommends the targets based on the comparison measures between other targets and users it's called collaborative filtering. A hybrid recommendation is based on a combination of content based and collaborative (Shaikh et al., 2017).

Content-based algorithms come with limitations of lack of diversified reviewer's interests, so the content fusion of reviewer behavior is suggested. It is implemented by building the correlation between the popularity of the reviewer's interest and the text and then finding the user preferences along with time utility and finally fusing the potential and user preferences to provide a recommendation list (Li & Wang, 2020).

In the collaborative filtering technique, the number of users increases the amount of work required by the system. The technique should be able to provide quality recommendations for complex problems. For complex problems, the preferred technique is item-based collaborative

filtering. The item-based technique uses indirect computing recommendations for the user from the relationship identified between different targets which is an output of the user-target matrix (Sarwar et al., 2001).

Xue and Zhang propose to calculate a new distance between the short and long text's similarity as a technique to identify the nearest neighbor set from the social network of the user and recommend the texts to the user's nearest neighbor set (Xue & Zhang, 2019).

A study on common recommendation techniques reveals that 55% of approaches are content based filtering, around 18% are collaborative filtering, and 16% are graph-based recommendations. Hybrid recommendations, stereotyping and item-centric recommendations are the other techniques that are applied (Beel et al., 2014).

A computer program designed to stimulate a conversation with a human user via textual methods is called a chatbot. Designing of chatbot includes using rules written in AIML or ChatScript provided by some good chat programs like a clever bot, do much more, Suzette and rosette (Haller & Rebedea, 2013).

Chapter 3: Problem Statement

“Credit cards are becoming less of a source of credit and more of a transactional platform. Many credit card users are switching from credit seekers to regular users of credit cards in place of cash. More users are using credit cards as a transactional medium due to convenience. Credit Cards provide greater convenience since users don’t need to carry large bundles of cash to make purchases. The credit card industry offers a platform where one can track their spending and consequently adjust their spending accordingly.” (*Credit Card Industry Analysis - Overview, Market Dynamics, Costs*, 2021).

In this project, *text analytics and sentiment analysis are performed on the reviews and tweets collected from websites and Twitter to develop tableau dashboards – which help to visualize the sentiments of users on using different credit cards based on credit card category, develop a recommendation engine for recommending the cards based on card categories and develop a simple chatbot which suggests best suitable credit cards* for usage on the transactional categories.

Chapter 4: Objectives of the Study

The scope of this project is to perform text analytics and sentiment analysis on reviews and tweets collected from the web and Twitter on credit card usage. NLP techniques and approaches are used for pre-processing the text and identifying the best algorithm which gives the highest accuracy.

Different techniques of feature extraction like TF-IDF, Text Blob, and Word2vec.

The objectives of the study are to develop,

- Tableau dashboard on sentiment analysis of credit card usage on transactional category and four cities of India.
- A recommendation engine to recommend similar credit cards based on the credit card categories.
- Simple Chat Bot, which suggests the best suitable credit card based on the user questions.

Chapter 5: Project Methodology

CRISP-DM framework has been used for this project as shown in the Figure No. 5.1.

The cross-industry standard process for data mining, known as CRISP-DM is an open standard process model that describes common approaches used by data mining experts. It is a widely used, analytics model (*Cross-Industry Standard Process for Data Mining - Wikipedia, 2022*).

“CRISP-DM breaks the process of data mining into six phases: Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, and Deployment. The sequences of phases are not strict and move back and forth between different phases as is always required. The arrows in the process diagram indicate the most important and frequent dependencies between phases. The outer circle in the diagram symbolizes the cyclic nature of data mining itself. A data mining process continues after a solution has been deployed. The lessons learned during the process can trigger new, often more focused business questions and subsequent data mining processes will benefit from the experiences of the previous one.”(*Cross-Industry Standard Process for Data Mining - Wikipedia, 2022*)

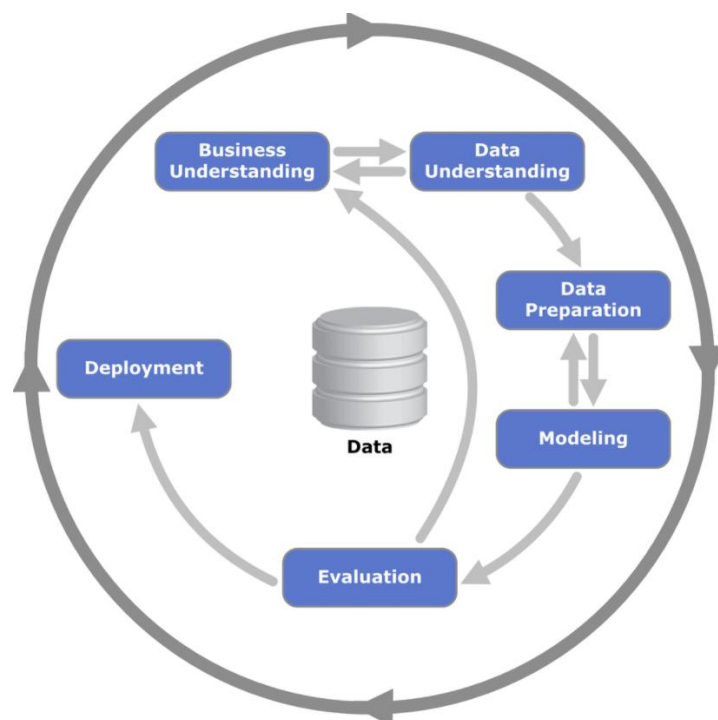


Figure No. 5.1 “CRISP-DM Framework”

Chapter 6: Business Understanding

With the help of emerging technologies, the finance sector has proved to be a significant part of our life. Fintech firms are attracted by the need to improve the financial services already offered by various institutions. Credit card plays an important role in the current transactional world. Credit card holders are switching from credit seekers to regular users of it, mainly due to transactional medium of convenience and better platform where users can track their spending (*Credit Card Industry Analysis - Overview, Market Dynamics, Costs*, 2021).

Cardholders can buy anything on credit within the limit and pay later concept. On high value purchases, users can convert the total amount into low-cost EMIs and repay easily over the period. Travel credit cards give users a special experience via priority check-in and complimentary lounge access. Some provide discounts on food at restaurants, and comprehensive travel insurance coverage. Learning how to use a credit card, and its repayment period will help to boost the credit score.

Sentiment analysis is one of the fastest growing research areas, which helps users to make better decisions on purchases through proper understanding and analysis of collective sentiments from the web and social media. It provides organizations to measure the impact of their social marketing strategies by identifying the public emotions towards the product or events associated with them (Shah, 2021).

It's a technique through which a piece of text is analyzed to determine the sentiment whether it's positive, negative, or neutral. It combines machine learning and natural language processing to achieve this. It focuses on the polarity of the text and goes beyond polarity to detect specific feelings and emotions, urgency (urgent, non-urgent), and even intentions (interested, not interested).

The scope of this project is to perform sentiment analysis on the reviews and tweets captured from the web and Twitter for credit cards. The commercial benefit of this project is the product will have dashboards that provide the sentiment details of credit cards at the transactional category level. A recommendation engine that recommends the user with credit cards that provides similar benefits to the existing card that which user has or does not have. Finally, a simple credit card bot that suggests a suitable credit card for users' transactional usage.

Chapter 7: Data Understanding

NLP architecture diagram Figure No. 7.1 used for this project is as shown below, data input taken here is the user reviews from the web from paisa bazaar and BankBazaar and tweets from Twitter.

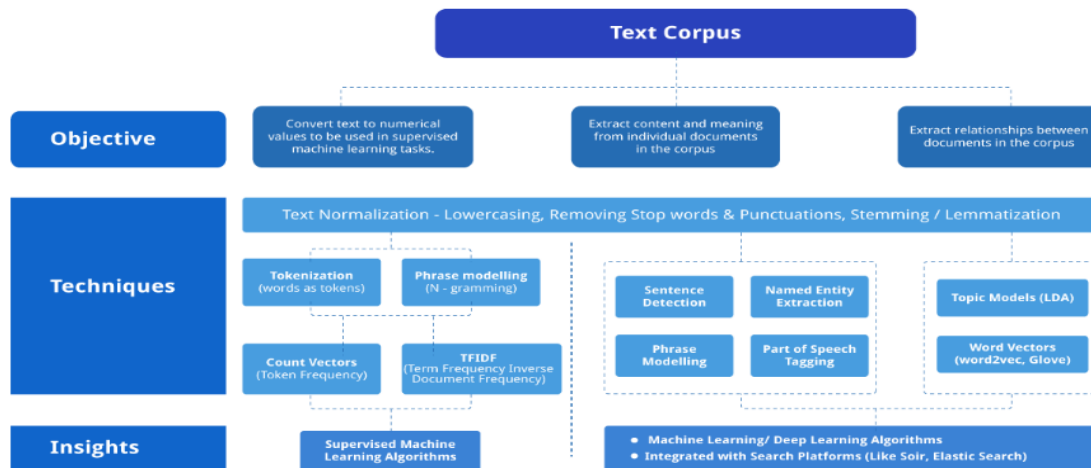


Figure No. 7.1 NLP Flow Diagram (Sankaran, 2017)

The data has been acquired from the website, Figure No. 7.2 using a trial version of the web scraping tool – Octoparse and using scraps and tweets from the Twitter API using R programming. Tweets from Twitter are acquired using the different parameters as shown in Figure No. 7.3.

Best Website for Checking Credit Score

★★★★★

Credit Report
My colleague in the office had suggested checking Paisabazaar's website for personal finance products. Where I found the option to 'Check Credit Score for Free'. I am extremely impressed with the service.

Siddharth Sharma
Posted on: Sep 24, 2021

Great Website for Checking Credit Score

★★★★★

Credit Report
I really liked your website. I have been planning to apply for a home loan for a long time so I downloaded my credit score and used the free tools to check my eligibility and the EMIs that I would have to pay on my loan.

Roopali
Posted on: Sep 23, 2021

Had a Good Experience

★★★★★

Credit Report
I wanted to apply for a credit card and someone at my office had told me to check my credit score before applying for it. A score above 750 is considered to be a good score for availing credit card. I checked my credit score from Paisabazaar.com and it was 780. Amazing service.

Megha
Posted on: Sep 23, 2021

Figure No. 7.2 Reviews text acquired from the web.

Bank Card Details	Geo Code	Product	Region
icici Credit Card	Bangalore - 12.97,77.59,150mi	icici	Bangalore
Axis Credit Card	Mumbai - 19.07,72.87,150mi	Axis	Mumbai
SBI Credit Card	Delhi - 28.70,77.10,150mi	SBI	Delhi
hdfc Credit Card	Kolkata - 22.57,88.36,150mi	hdfc	Kolkata

Figure No. 7.3 Parameters used to acquire tweets from Twitter.

Title	View	View1	shortprofile	shortprofile2
Good card with travel benefits	HDFC Bank Credit Card	I got a travel-specific HDFC credit card in 2019. L	Mrityunjay Abhivyakti Posted on: Jun 2, 2021	Mrityunjay Abhivyakti
Good benefits but less rewards	State Bank of India Credit Card	I have been using SBI Credit Card since 2019. I lik	Mangesh Kumar Posted on: Jun 2, 2021	Mangesh Kumar
Exciting perks and benefits	ICICI Bank Credit Card	One thing is for sure that Paisabazaar has quality	Ojasvini Posted on: Jun 2, 2021	Ojasvini
HDFC Moneyback Credit Card	HDFC Bank Credit Card	This was my first credit card and I still have it wit	Rahul Kumar Arya Posted on: May 25, 2021	Rahul Kumar Arya

Figure No. 7.4 Preprocessing data extract from Web.

ttext	date	isretweet	retweetcount	favoritecount	score	product	region	country	duplicate
OnePlus 9RT in stock	19-01-2022 02:04	FALSE	0	0	0	SBI	Delhi	India	FALSE
@AxisBankSupport i haven't applied for Flipkart	19-01-2022 02:44	FALSE	0	0	0	Axis	Delhi	India	FALSE
@HDFC_Bank My loan application and credit ca	19-01-2022 03:34	FALSE	0	0	0	hdfc	Delhi	India	FALSE
Hey @AxisBank , i didn't received my credit car	19-01-2022 03:53	FALSE	0	1	-1	Axis	Delhi	India	FALSE
@TechnoFino Bro I find a term in my hdfc millinea credit card statement. 1 RP Premium LTF. What is this mean? Is m	19-01-2022 04:03	FALSE	0	0	0	hdfc	Delhi	India	FALSE
@HDFC_Bank why have you been charging me	19-01-2022 04:24	FALSE	0	0	0	hdfc	Bangalore	India	FALSE

Figure No. 7.5 Preprocessing data from Twitter.

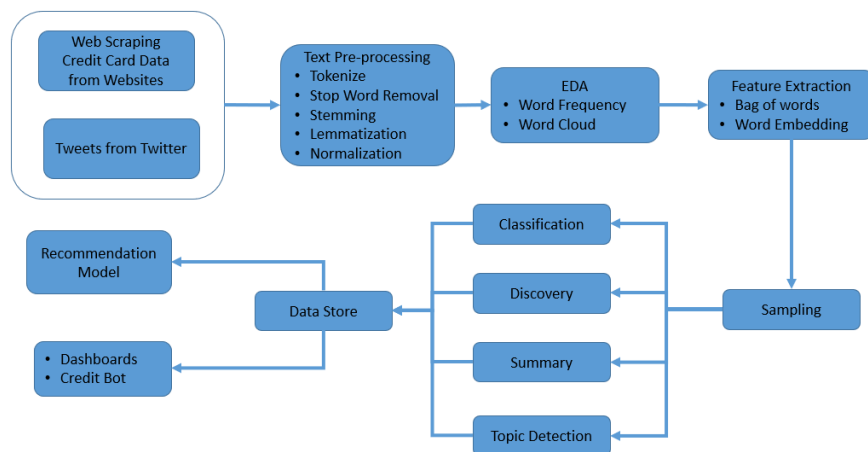


Figure No. 7.6 Data Pipeline.

The first step is to extract reviews and tweets for the web and Twitter. In text pre-processing, various preprocessing activities on the text are done and then the Lexicon method of scoring is performed. EDA describes the frequency of words, repeated words, and so on. Once EDA is done, feature extraction is performed on the unstructured text data into machine readable format and numbers using a bag of words, TF-IDF, and word embedding. Sampling and then various classifiers, discovery, summary, and topic detection are done which gives the sentiment of texts as positive, negative, and neutral on the reviews and tweets.

Chapter 8: Data Preparation

Data processing is an important step to make data ready for the modeling process. The following steps were involved for text preprocessing:

Basic feature extraction:

The number of words: Number of words in each comment extracted as shown in Figure No. 8.1.

	comment	word_count
0	On my first stay with the Trident Group, I was...	15
1	From shopping at the supermarkets to watching ...	16
2	Have been using this card for over 2 years now...	20
3	This card is indeed the one that offers the mo...	18
4	The Valueback deals on fuels are the perfect b...	12

Figure No. 8.1 The number of words

The number of characters: The number of characters in each comment includes spaces, which can be removed if not required as shown in Figure No. 8.2.

	comment	char_count
0	On my first stay with the Trident Group, I was...	79.0
1	From shopping at the supermarkets to watching ...	98.0
2	Have been using this card for over 2 years now...	96.0
3	This card is indeed the one that offers the mo...	101.0
4	The Valueback deals on fuels are the perfect b...	66.0

Figure No. 8.2 Number of characters.

The number of stop words: There are some most common words in the language. Examples are he, she, have, are, is, etc. which are already captured in the corpus. Usually, they don't add much meaning to the sentence as shown in Figure No 8.3. Library package stopwords imported from NLTK.

	comment	stopwords
0	On my first stay with the Trident Group, I was...	5
1	From shopping at the supermarkets to watching ...	7
2	Have been using this card for over 2 years now...	10
3	This card is indeed the one that offers the mo...	7
4	The Valueback deals on fuels are the perfect b...	5

Figure No. 8.3 Number of stop words.

The number of special characters: Special characters like hashtags are also an interesting feature of the extract, which helps to get additional information from the text data as shown in Figure No. 8.4.

	comment	hashtags
0	On my first stay with the Trident Group, I was...	0
1	From shopping at the supermarkets to watching ...	0
2	Have been using this card for over 2 years now...	0
3	This card is indeed the one that offers the mo...	0
4	The Valueback deals on fuels are the perfect b...	0

Figure No. 8.4 The number of special characters.

The number of numeric: Calculating the number of numeric is a similar exercise to calculating the number of words as shown in Figure No 8.5.

	comment	numerics
0	On my first stay with the Trident Group, I was...	1
1	From shopping at the supermarkets to watching ...	0
2	Have been using this card for over 2 years now...	1
3	This card is indeed the one that offers the mo...	0
4	The Valueback deals on fuels are the perfect b...	0

Figure No. 8.5 Count of numeric values.

Text processing:

To get better features, we need to clean the text and features extracted. Basic text pre-processing steps include:

Lowercase: Sentences are converted in lowercase, which helps to eliminate multiple copies of the same words in the data as shown in Figure No. 8.6.

```
0   on my first stay with the trident group, i was...
1   from shopping at the supermarkets to watching ...
2   have been using this card for over 2 years now...
3   this card is indeed the one that offers the mo...
4   the valueback deals on fuels are the perfect b...
Name: comment, dtype: object
```

Figure No. 8.6 Lower Case.

Removing Punctuations: Punctuations in the text data are removed, as they won't add any meaning to the text as shown in Figure No. 8.7.

```
0   on my first stay with the trident group i was ...
1   from shopping at the supermarkets to watching ...
2   have been using this card for over 2 years now...
3   this card is indeed the one that offers the mo...
4   the valueback deals on fuels are the perfect b...
Name: comment, dtype: object
```

Figure No. 8.7 Removing Punctuations.

Removing Stopwords: Commonly occurring words should be removed from the data. Predefined libraries are used. Some examples of stop words are I, me, myself, etc as shown in Figure No. 8.8.

```
0   first stay trident group rewarded 1500 bonus p...
1   shopping supermarkets watching movies spending...
2       using card 2 years benefits rewards good
3   card indeed one offers rewarding fuel saving b...
4       valueback deals fuels perfect benefit card
Name: comment, dtype: object
```

Figure No. 8.8 Removing Stop Words.

Spelling Correction: Spelling mistakes are common in the text collected from the websites which need to be corrected before we build the model. Text Blob library of python is used to correct spellings as shown in Figure No. 8.9.

```
0   first stay strident group rewarded 1500 bonus ...
1   shopping supermarket watching moves spending s...
2       using card 2 years benefits rewards good
3   card indeed one offers rewarding fuel saving b...
4       valueback deals feels perfect benefit card
Name: comment, dtype: object
```

Figure No. 8.9 Spelling Correction.

Tokenization: Tokens are building blocks of NLP. It is breaking the text into words, sentences called tokens. Token helps in understanding the context or developing the model and interpreting the meaning by analyzing the sequences of the words as shown in Figure No. 8.10.

```
0    [first, stay, trident, group, rewarded, 1500, ...
1    [shopping, supermarkets, watching, movies, spe...
2    [using, card, 2, years, benefits, rewards, good]
3    [card, indeed, one, offers, rewarding, fuel, s...
4    [valueback, deals, fuels, perfect, benefit, card]
Name: comment, dtype: object
```

Figure No. 8.10 Tokenization

Stemming: Its normalization technique of words i.e., the process of reducing a word into its word stem. For example, a stem of the word “studying” is study, as shown in Figure No. 8.11.

```
0    first stay trident group reward 1500 bonu point
1    shop supermarket watch movi spend card reward
2    use card year benefit reward good
3    card inde offer reward fuel save benefit card ...
4    valueback deal fuel perfect benefit card
Name: comments, dtype: object
```

Figure No. 8.11 Stemming

Lemmatization: It considers vocabulary and morphological analysis of words. Here the algorithm links the dictionaries in the form of a lemma which is a canonical form of a word as shown in Figure No. 8.12.

```
0    first stay trident group rewarded 1500 bonus p...
1    shopping supermarket watching movie spending c...
2    using card year benefit reward good
3    card indeed offer rewarding fuel saving benefi...
4    valueback deal fuel perfect benefit card
Name: comments, dtype: object
```

Figure No. 8.12 Lemmatization

N-grams: Combination of multiple words used together, they capture language structure i.e., most likely word or letter which would follow the given one. It's called unigram when N =1, bigram when N=2, trigram when N=3, and so on.

```
[WordList(['first', 'stay']),
 WordList(['stay', 'trident']),
 WordList(['trident', 'group']),
 WordList(['group', 'rewarded']),
 WordList(['rewarded', '1500']),
 WordList(['1500', 'bonus']),
 WordList(['bonus', 'point'])]
```

Figure No. 8.13 N-grams

Term Frequency: Measures how frequently a term occurs in a document. The number of times term t appears in a document / Total number of terms in the document.

Inverse Document Frequency: Measure how important is the term. $\log_e(\text{Total number of documents} / \text{Number of documents with term } t \text{ in it})$

	words	tf	idf	tfidf
0	shopping	1	2.221439	2.221439
1	supermarket	1	6.345636	6.345636
2	watching	1	8.050384	8.050384
3	movie	1	3.726252	3.726252
4	spending	1	3.003739	3.003739
5	card	1	0.293975	0.293975
6	rewarding	1	3.573048	3.573048

Figure No. 8.14 TF IDF

Bag of Words: Represents text which describes the presence of words within text data extracted. Two similar text fields will contain similar kinds of words and therefore have a similar bag of words. From this text, we learn about the meaning of the document as shown in Figure No. 8.15.

```
['sbicardconnect',  
'stop',  
'loot',  
'peopl',  
'genuin',  
'pathet',  
'experi',  
'one',  
'get',  
'trap']
```

Figure No. 8.15 Bag of Words

Most Commonly Occurring Words: Commonly occurring words in text data as shown in Figure No. 8.16.

```
[('card', 4832),  
(('credit', 2690),  
(('reward', 2459),  
(('sbi', 1225),  
(('get', 1097),  
(('point', 1053),  
(('use', 1023),  
(('bank', 1000),  
(('shop', 741),  
(('fuel', 583),  
(('got', 564),  
(('offer', 520),  
(('good', 496),  
(('limit', 493),  
(('hdfc', 480),  
(('icici', 473),  
(('inn', 461),  
(('benefit', 459),  
(('hdfcbank', 435),  
(('spend', 429)]
```

Figure No. 8.16 Most Common Words

Word Frequency: Indicates the number of times each token occurs in a text. The top 5 commonly used words from the data extract are card, credit, reward, sbi, and get as shown in Figure No. 8.17.

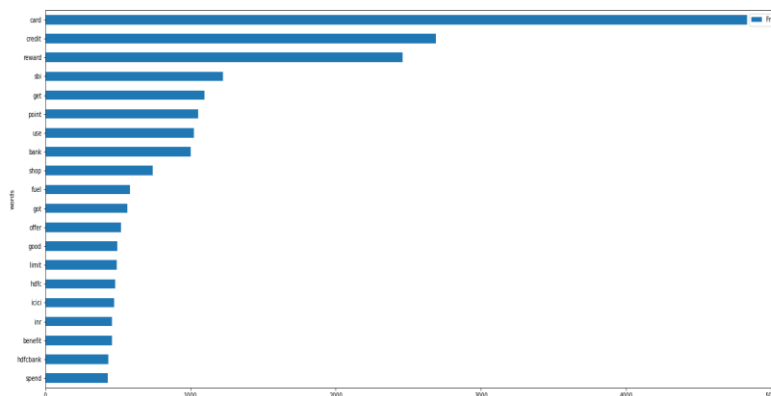


Figure No. 8.17 Word Frequency

Word Cloud: Words used in text or subject within an image or picture and the size of each word indicates its frequency as shown in Figure No. 8.18.



Figure No. 8.18 Word Cloud

Word Frequency Distribution: Figure No. 8.19 shows the distribution of word frequency. The top 5 commonly used words from the data extract are card, credit, reward, sbi, and get as shown in Figure No. 8.19

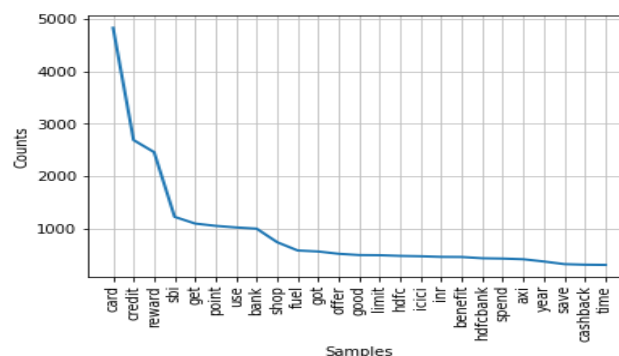


Figure No. 8.19 Word Frequency Distribution

Data Preparation steps for Recommendation Engine.

Step 1. Collect the data which has the polarity value from previously processed data.

Step 2. Filter the rows which have sentiment values of “Neutral” or “Negative”

Step 3. Create a feature with the concatenation of two features “Credit Card” and “Card Category” as shown in Figure No. 8.20.

serial_number	credit_card	reviews	user	comment	polarity	polarity_type	card_category	names	userid	credit_card_category	
0	0	HDFC Bank Credit Card	travel-specific HDFC credit card 2019. Luckily...	Mrityunjay Abhivyakti	travel-specif hdfc credit card 2019. luckily, ...	0.400000	Positive	reward	Mrityunjay Abhivyakti	0	HDFC Bank Credit Card - reward
1	1	State Bank of India Credit Card	have been using Credit Card since 2019. like s...	Mangesh Kumar	have been use credit card sinc 2019. like shop...	0.431667	Positive	shopping	Mangesh Kumar	1	State Bank of India Credit Card - shopping
2	2	ICICI Bank Credit Card	thing sure that Paisabazaar quality profession...	Ojasvini	thing sure that paisabazaar qualiti profession...	0.550000	Positive	general	Ojasvini	2	ICICI Bank Credit Card - general
3	3	HDFC Bank Credit Card	This first credit card still have with because...	Rahul Kumar Arya	thi first credit card still have with becaus s...	0.316667	Positive	reward	Rahul Kumar Arya	3	HDFC Bank Credit Card - reward
4	6	State Bank of India Credit Card	Simply Click good card online shopping reward...	Ritu Kushwaha	simpli click good card onlin shopping rewards...	0.700000	Positive	shopping	Ritu Kushwaha	4	American Express Credit Card - reward

Figure No. 8.20 Preprocessed Data for Recommendation Engine.

Step 4. Reduce the dataset to the required features as shown in Figure No. 8.21.

	userid	credit_card_category	polarity
0	0	HDFC Bank Credit Card - reward	0.400000
1	1	State Bank of India Credit Card - shopping	0.431667
2	2	ICICI Bank Credit Card - general	0.550000
3	3	HDFC Bank Credit Card - reward	0.316667
4	4	American Express Credit Card - reward	0.700000

Figure No. 8.21 Dataset with reduced features for Recommendation Engine.

Step 5. Binning the dataset based on Polarity values as shown in Figure No. 8.22 and Figure No. 8.23.

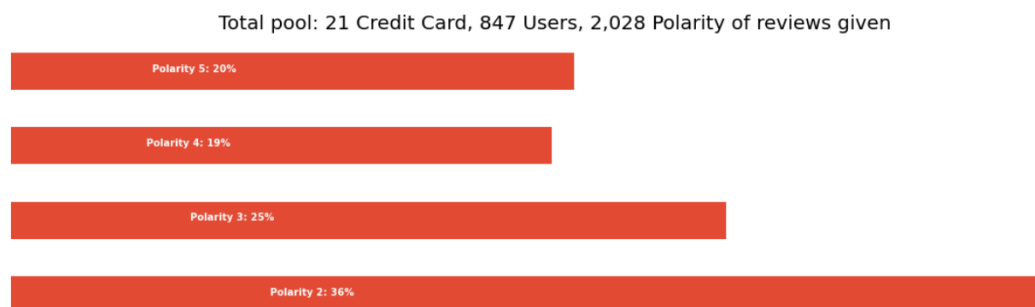


Figure No. 8.22 Dataset with binned values of Polarity.

	userid	credit_card_category	polarity_bin
0	0	HDFC Bank Credit Card - reward	2
1	1	State Bank of India Credit Card - shopping	3
2	2	ICICI Bank Credit Card - general	3
3	3	HDFC Bank Credit Card - reward	2
4	4	American Express Credit Card - reward	4

Figure No. 8.23 Dataset for Recommendation Engine

Data Preparation for Simple Chat-Bot

Dataset is used to train the chatbot which contains responses, intents, and patterns as shown in Figure No. 8.24.

```
{
  "intents": [
    {
      "tag": "greeting",
      "patterns": ["Hi there", "How are you", "Is anyone there?", "Hey", "Hola", "Hello", "Good day"],
      "responses": ["Hello - I am Credit Card Bot, How Can I help you?"],
      "context": [""]
    },
    {
      "tag": "goodbye",
      "patterns": ["Bye", "See you later", "Goodbye", "Nice chatting to you, bye", "Till next time"],
      "responses": ["See you!", "Have a nice day", "Bye! Come back again soon."],
      "context": [""]
    },
    {
      "tag": "thanks",
      "patterns": ["Thanks", "Thank you", "That's helpful", "Awesome, thanks", "Thanks for helping me"],
      "responses": ["Happy to help!", "Any time!", "My pleasure"],
      "context": [""]
    },
    {
      "tag": "noanswer",
      "patterns": [" "],
      "responses": ["Sorry, can't understand you", "Please give me more info", "Not sure I understand"],
      "context": [""]
    },
    {
      "tag": "card",
      "patterns": ["Credit Card", "I want Credit Card", "Card"],
      "responses": ["Please type the approx montly spend income 10000 or 20000 or 30000 or 40000."],
      "context": [""]
    },
    {
      "tag": "category",
      "patterns": ["10000", "20000", "30000", "40000"],
      "responses": ["Choose Credit Card category to like to spend Fuel, Shopping, Reward, Cashback, Lifestyle, Travel"],
      "context": [""]
    }
  ]
}
```

Figure No. 8.24 Dataset .json file used for training Chat Bot.

Chapter 9: Modeling

Preprocessed data discussed in the previous section was fed into multiple models to the polarity value for the sentiments. A classification algorithm is a supervised learning technique used to classify classification observations based on a training dataset set. The classification technique used based on different approaches is shown below.

Lexicon-Based Approach:

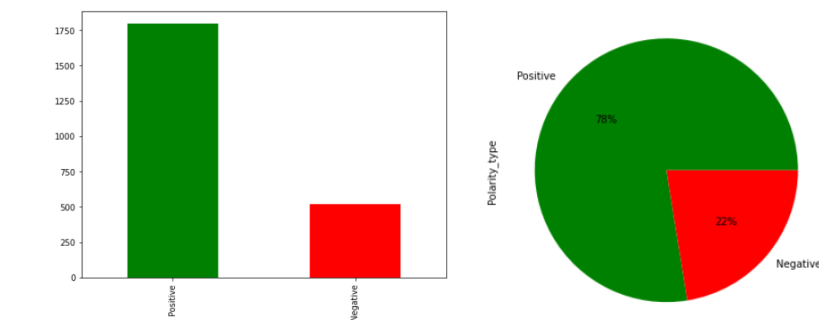


Figure No. 9.1 EDA of Lexicon Approach

Dataset has 1796 positive texts which are 78% and 520 negative texts 22% of which are used for training and testing the model as shown in Figure No. 9.1 and code as in Figure No. 9.2.

```
lexicons = {}
records = lex_file.readlines()
for record in records:
    #print(record) # Line contains newline character
    #print(record.rstrip('\n').split(",")) - to remove new line character
    lexicons[record.rstrip('\n').split(",")[0]] = int(record.rstrip('\n').split(",")[1])
print(lexicons)
#Lexicons["abandon"]

strength = []
for Comments in word_list:
    score = 0
    for word in Comments:
        if word in (lexicons):
            score = score + lexicons[word]
    strength.append(score)

senti_matrix = pd.DataFrame(strength, corpus)

def clean_comment_length(UserComments):
    letters_only = re.sub("[^a-zA-Z]", " ", UserComments)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    return(len(meaningful_words) )

def comment_to_words(UserComments):
    letters_only = re.sub("[^a-zA-Z]", " ", UserComments)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    return( " ".join( meaningful_words ))

dflex['clean_comment_length']=dflex['UserComments'].apply(lambda x: clean_comment_length(x))
dflex['comment_length']=dflex['UserComments'].apply(lambda x: clean_comment_length(x))
train,test = train_test_split(dflex,test_size=0.25,random_state=42)

train_clean_comment=[]
for comment in train['clean_comment']:
    train_clean_comment.append(comment)
test_clean_comments=[]
for comment in test['clean_comment']:
    test_clean_comment.append(comment)

from sklearn.feature_extraction.text import CountVectorizer
v = CountVectorizer(analyzer = "word")
train_features= v.fit_transform(train_clean_comment)
test_features=v.transform(test_clean_comment)
```

Figure No. 9.2 Models used for Lexicon Approach

TextBlob Sentiment Analysis:

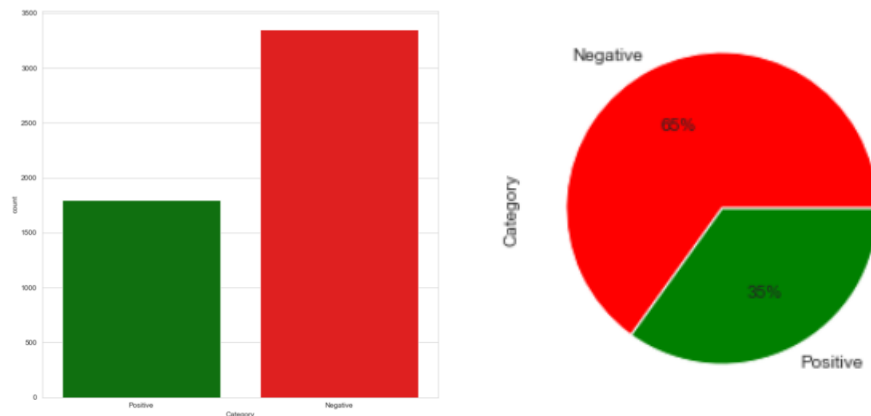


Figure No. 9.3 EDA of TextBlob Sentiment Analysis

Dataset has 1796 positive texts which are 36% and 3348 negative texts which are 65% of the text extracts which is used for modeling as shown in Figure No. 9.3 and code as in Figure 9.4.

```
sentiment_text = [TextBlob(Comments) for Comments in corpus]
print(sentiment_text[10].polarity)
print(sentiment_text[10])

0.0
like amaz reward card sbi

sentiments = [[Comments.sentiment.polarity, str(Comments)] for Comments in sentiment_text]
sentiment_df = pd.DataFrame(sentiments, columns=["Polarity", "UserComments"])
sentiment_df.sort_values(by='Polarity', ascending=False)

def clean_comment_length(UserComments):
    letters_only = re.sub("[^a-zA-Z]", " ", UserComments)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    return(len(meaningful_words) )

def comment_to_words(UserComments):
    letters_only = re.sub("[^a-zA-Z]", " ", UserComments)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    return( " ".join( meaningful_words ))

dflex['clean_comment']=dflex['UserComments'].apply(lambda x: comment_to_words(x))
dflex['comment_length']=dflex['UserComments'].apply(lambda x: clean_comment_length(x))
train,test = train_test_split(dflex,test_size=0.25,random_state=42)

train_clean_comment=[]
for comment in train['clean_comment']:
    train_clean_comment.append(comment)
test_clean_comment=[]
for comment in test['clean_comment']:
    test_clean_comment.append(comment)

from sklearn.feature_extraction.text import CountVectorizer
v = CountVectorizer(analyzer = "word")
train_features= v.fit_transform(train_clean_comment)
test_features=v.transform(test_clean_comment)
```

Figure No. 9.4 Models for TextBlob Sentiment Analysis

VADER Sentiment Analysis:

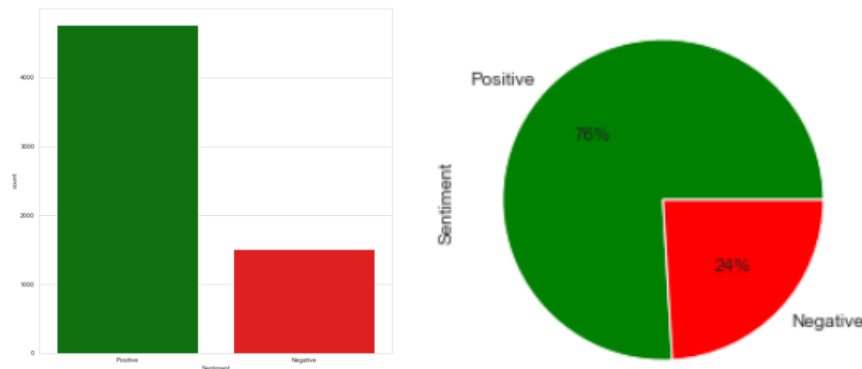


Figure No. 9.5 EDA of VADER Sentiment Analysis

Dataset has 4760 positive text which is 76% and 1510 negative text which is 24% of the extract data used in modeling as shown in Figure No. 9.5 and code as in Figure No. 9.6.

```
from pandas import DataFrame
df_clean = DataFrame(corpus, columns=['comment'])

from vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer
analyser = SentimentIntensityAnalyzer()

def sentiment_analyzer_scores(sentence):
    score = analyser.polarity_scores(sentence)
    print("{:-<40} {}".format(sentence, str(score)))

df_clean_new = df_clean.comment[0:5]

sent_analyser = SentimentIntensityAnalyzer()
def sentiment(text):
    return (sent_analyser.polarity_scores(text)["compound"])

def senti(df):
    if df['Polarity'] >= 0.3:
        val = "Positive"
    elif df['Polarity'] <= -0.25:
        val = "Negative"
    else:
        val = "Negative"
    return val

def clean_comment_length(UserComments):
    letters_only = re.sub("[^a-zA-Z]", " ", UserComments)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    return(len(meaningful_words) )

def comment_to_words(UserComments):
    letters_only = re.sub("[^a-zA-Z]", " ", UserComments)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    return( " ".join( meaningful_words ) )

dflex['clean_comment']=dflex['UserComments'].apply(lambda x: comment_to_words(x))
dflex['comment_length']=dflex['UserComments'].apply(lambda x: clean_comment_length(x))
train,test = train_test_split(dflex,test_size=0.25,random_state=42)

train_clean_comment=[]
for comment in train['clean_comment']:
    train_clean_comment.append(comment)
test_clean_comment=[]
for comment in test['clean_comment']:
    test_clean_comment.append(comment)

from sklearn.feature_extraction.text import CountVectorizer
v = CountVectorizer(analyzer = "word")
train_features= v.fit_transform(train_clean_comment)
test_features=v.transform(test_clean_comment)
```

Figure No. 9.6 Models for VADER Sentiment Analysis

TF-IDF Vectorizer:

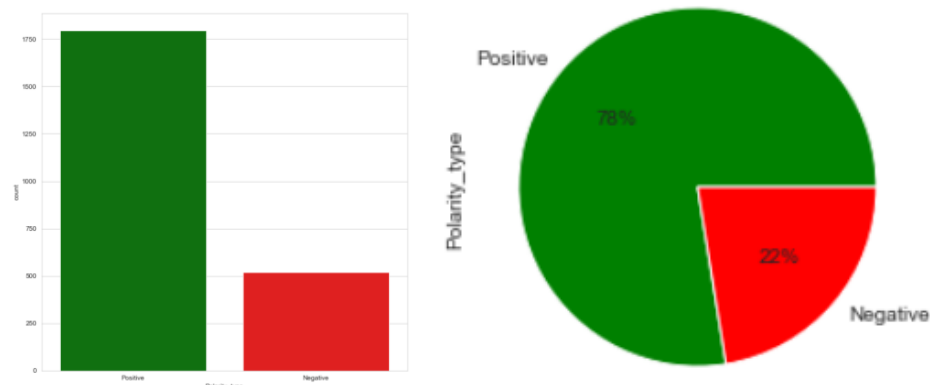


Figure No. 9.7 EDA of TF-IDF Vectorizer

Dataset has 1796 positive text which is 78% and 520 negative text which is 22% of the extracted data for modeling as shown in Figure No. 9.7 and code as in Figure No. 9.7.

```

def clean_comment_length(UserComments):
    letters_only = re.sub("[^a-zA-Z]", " ", UserComments)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    return(len(meaningful_words))

def comment_to_words(UserComments):
    letters_only = re.sub("[^a-zA-Z]", " ", UserComments)
    words = letters_only.lower().split()
    stops = set(stopwords.words("english"))
    meaningful_words = [w for w in words if not w in stops]
    return(" ".join(meaningful_words))

dfc['clean_comment']=dfc['UserComments'].apply(lambda x: comment_to_words(x))
dfc['comment_length']=dfc['UserComments'].apply(lambda x: clean_comment_length(x))
train,test = train_test_split(dfc,test_size=0.25,random_state=42)

train_clean_comment=[]
for comment in train['clean_comment']:
    train_clean_comment.append(comment)
test_clean_comment=[]
for comment in test['clean_comment']:
    test_clean_comment.append(comment)

from sklearn.feature_extraction.text import TfidfVectorizer
v = TfidfVectorizer(max_features = 4500)
train_features= v.fit_transform(train_clean_comment)
test_features=v.transform(test_clean_comment)
    
```

Figure No. 9.8 Models for TF-IDF Vectorizer

Fine Tuning with BERT:

Google has open-sourced technique BERT, with this model can train as transfer learning. It is a deep learning bidirectional unsupervised language representation. A plain text corpus is used to train the model and the output is as shown in Figure No. 9.9.

```
| # Get the lists of sentences and their labels.
sentences = dpl_data.UserComments.values
labels = dpl_data.Label.values

| from transformers import BertTokenizer

# Load the BERT tokenizer.
print('Loading BERT tokenizer...')
tokenizer = BertTokenizer.from_pretrained('bert-base-uncased', do_lower_case=True)

Loading BERT tokenizer...

| # Print the original sentence.
print(' Original: ', sentences[0])

# Print the sentence split into tokens.
print('Tokenized: ', tokenizer.tokenize(sentences[0]))

# Print the sentence mapped to token ids.
print('Token IDs: ', tokenizer.convert_tokens_to_ids(tokenizer.tokenize(sentences[0])))

| from torch.utils.data import TensorDataset, random_split

# Combine the training inputs into a TensorDataset.
dataset = TensorDataset(input_ids, attention_masks, labels)

# Create a 90-10 train-validation split.

# Calculate the number of samples to include in each set.
train_size = int(0.9 * len(dataset))
val_size = len(dataset) - train_size

# Divide the dataset by randomly selecting samples.
train_dataset, val_dataset = random_split(dataset, [train_size, val_size])

print('{:>5,} training samples'.format(train_size))
print('{:>5,} validation samples'.format(val_size))

1,233 training samples
 138 validation samples

| from torch.utils.data import DataLoader, RandomSampler, SequentialSampler

# The DataLoader needs to know our batch size for training, so we specify it
# here. For fine-tuning BERT on a specific task, the authors recommend a batch
# size of 16 or 32.
batch_size = 32

# Create the DataLoaders for our training and validation sets.
# We'll take training samples in random order.
train_dataloader = DataLoader(
    train_dataset, # The training samples.
    sampler = RandomSampler(train_dataset), # Select batches randomly
    batch_size = batch_size # Trains with this batch size.
)

# For validation the order doesn't matter, so we'll just read them sequentially.
validation_dataloader = DataLoader(
    val_dataset, # The validation samples.
    sampler = SequentialSampler(val_dataset), # Pull out batches sequentially.
    batch_size = batch_size # Evaluate with this batch size.
)

| from transformers import BertForSequenceClassification, AdamW, BertConfig

# Load BertForSequenceClassification, the pretrained BERT model with a single
# linear classification layer on top.
model = BertForSequenceClassification.from_pretrained(
    "bert-base-uncased", # Use the 12-layer BERT model, with an uncased vocab.
    num_labels = 2, # The number of output labels--2 for binary classification.
                    # You can increase this for multi-class tasks.
    output_attentions = False, # Whether the model returns attentions weights.
    output_hidden_states = False, # Whether the model returns all hidden-states.
)

# Tell pytorch to run this model on the GPU.
#model.cuda()
```

```

# Get all of the model's parameters as a list of tuples.
params = list(model.named_parameters())

print('The BERT model has {} different named parameters.\n'.format(len(params)))

print('==== Embedding Layer ====')

for p in params[0:5]:
    print("{}:{<55} {:>12}".format(p[0], str(tuple(p[1].size()))))

print('\n==== First Transformer ====')

for p in params[5:21]:
    print("{}:{<55} {:>12}".format(p[0], str(tuple(p[1].size()))))

print('\n==== Output Layer ====')

for p in params[-4:]:
    print("{}:{<55} {:>12}".format(p[0], str(tuple(p[1].size()))))

The BERT model has 201 different named parameters.

==== Embedding Layer ====

bert.embeddings.word_embeddings.weight          (30522, 768)
bert.embeddings.position_embeddings.weight       (512, 768)
bert.embeddings.token_type_embeddings.weight     (2, 768)
bert.embeddings.LayerNorm.weight                (768,)
bert.embeddings.LayerNorm.bias                  (768,)

==== First Transformer ====

bert.encoder.layer.0.attention.self.query.weight (768, 768)
bert.encoder.layer.0.attention.self.query.bias  (768,)
bert.encoder.layer.0.attention.self.key.weight  (768, 768)
bert.encoder.layer.0.attention.self.key.bias    (768,)
bert.encoder.layer.0.attention.self.value.weight (768, 768)
bert.encoder.layer.0.attention.self.value.bias  (768,)
bert.encoder.layer.0.attention.output.dense.weight (768, 768)
bert.encoder.layer.0.attention.output.dense.bias (768,)
bert.encoder.layer.0.attention.output.LayerNorm.weight (768,)
bert.encoder.layer.0.attention.output.LayerNorm.bias (768,)
bert.encoder.layer.0.intermediate.dense.weight (3072, 768)
bert.encoder.layer.0.intermediate.dense.bias    (3072,)
bert.encoder.layer.0.output.dense.weight        (768, 3072)
bert.encoder.layer.0.output.dense.bias          (768,)
bert.encoder.layer.0.output.LayerNorm.weight    (768,)
bert.encoder.layer.0.output.LayerNorm.bias      (768,)

==== Output Layer ====

bert.pooler.dense.weight          (768, 768)
bert.pooler.dense.bias            (768,)
classifier.weight                  (2, 768)
classifier.bias                    (2,)

```

Figure No. 9.9 BERT Model

Recommendation Engine:

KNN-based collaborative filtering model relies on the similarity of item features data distribution without making any assumptions. Distance is calculated between the target/label and every other item label/ target within the dataset. Top k items are returned based on ranks calculated between the distances. Code as shown in Figure No. 9.10.

```

data['credit_card_category']
len(set(data['credit_card_category']))
pivot_table = data.pivot_table(index='credit_card_category', columns='userid', values='polarity_bin' ).fillna(0)
pivot_table.shape

from scipy.sparse import csr_matrix
feature_matrix = csr_matrix(pivot_table.values)
feature_matrix

from sklearn.neighbors import NearestNeighbors
knn_model = NearestNeighbors(metric='cosine', algorithm='brute', )
knn_model.fit(feature_matrix)
query_index = np.random.choice(pivot_table.shape[0])
query_index
query_id = np.random.choice(pivot_table.shape[1])
query_id

pivot_table.iloc[16:]
pivot_table.iloc[16,:].values.reshape(1,-1)

distances, indices = knn_model.kneighbors(pivot_table.iloc[query_index,:].values.reshape(1,-1),n_neighbors=5)

distances
indices.flatten()
pivot_table.index[16]
pivot_table.iloc[indices.flatten()].index

for i in range(0,len(distances.flatten())):
    if i == 0:
        print('Recommendations for {}'.format(pivot_table.index[query_index]))
    else:
        print('{} : {}, with a distance of {}'.format(i,pivot_table.index[indices.flatten()[i]], distances.flatten()[i]))

```

Figure No. 9.10 KNN Based Collaborative Model

Chat Bot Using NLTK and Kera's

A chatbot is software that can communicate and perform like humans. Two basic types of chatbot models based on how they are built, are retrieval and generative models. The retrieval-based chatbot uses predefined input patterns and responses, then uses a heuristic approach to select the appropriate response. Code as shown in Figure No. 9.11.

```
# Create model - 3 layers. First layer 128 neurons, second layer 64 neurons and 3rd output layer contains number of neurons
# equal to number of intents to predict output intent with softmax
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))

# Compile model. Stochastic gradient descent with Nesterov accelerated gradient gives good results for this model
sgd = SGD(lr=0.01, decay=1e-6, momentum=0.9, nesterov=True)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])

#fitting and saving the model
hist = model.fit(np.array(train_x), np.array(train_y), epochs=100, batch_size=5, verbose=1)
model.save('C:\\Reva\\BA06\\2nd Year\\capstone_project\\chatbot\\chatbot-python-project-data-codes\\chatbot_model.h5', hist)

print("model created")
```

Figure No. 9.11 Bot using NLTK and Keras.

Chapter 10: Model Evaluation

Machine Learning model approaches used are Lexicon with AFFIN vocabulary, classifiers with Text Blob Polarity, VADER Sentiment, TF-IDF Vectorizer, and Transfer Learning using fine-tuned BERT Model are compared. Observed performance of classifiers and deep learning models, it's important to have accurate labeling for all comments. Accuracies of models are shown in Table No 10.1 and Figure No. 10.2.

S.N.	Approach	Accuracy	Classifier / Model with Best Result
1	Text Blob Polarity	98%	Decision Tree Classifier
2	Fine Tuning with BERT	97%	Transfer Learning
3	Word Embedding TF-IDF Vectorizer	94%	Gradient Boosting Classifier
4	Lexicon Vocabulary	93%	Gradient Boosting Classifier
5	VADER Sentiment	90%	Random Forest Classifier

Table No 10.1 Accuracies of Modelling.

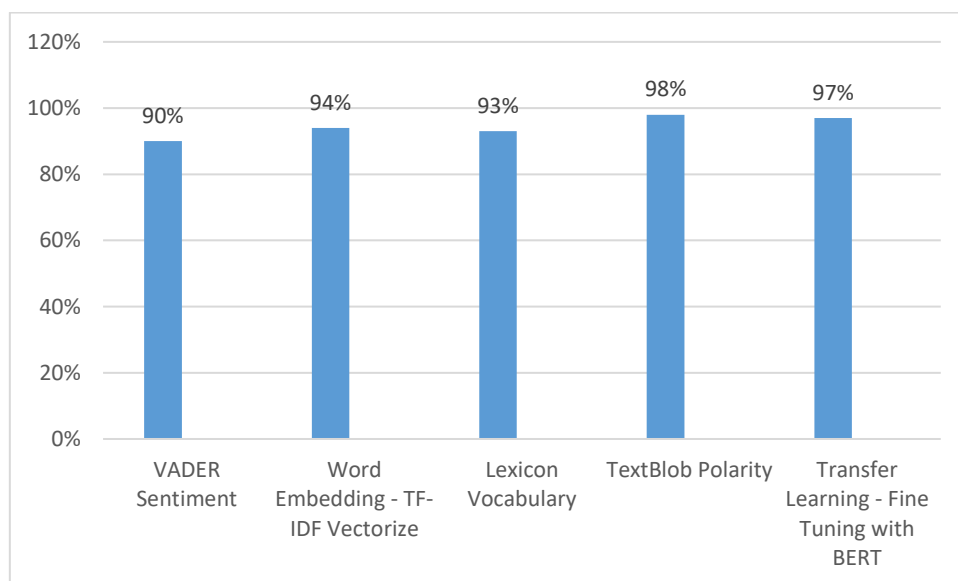


Figure No. 10.2 Comparison of Accuracies

Text Blob Polarity with Decision Tree Classifier has given the highest accuracy of 98% for the sentiment analysis as shown in Figure No. 10.2 for TextBlob models comparison, Table No 10.2 shows the classifier model's results and Confusion Matrix as shown in Figure No. 10.3. There are various factors affecting model performance for other approaches, especially for deep learning techniques which need to have more optimized data.

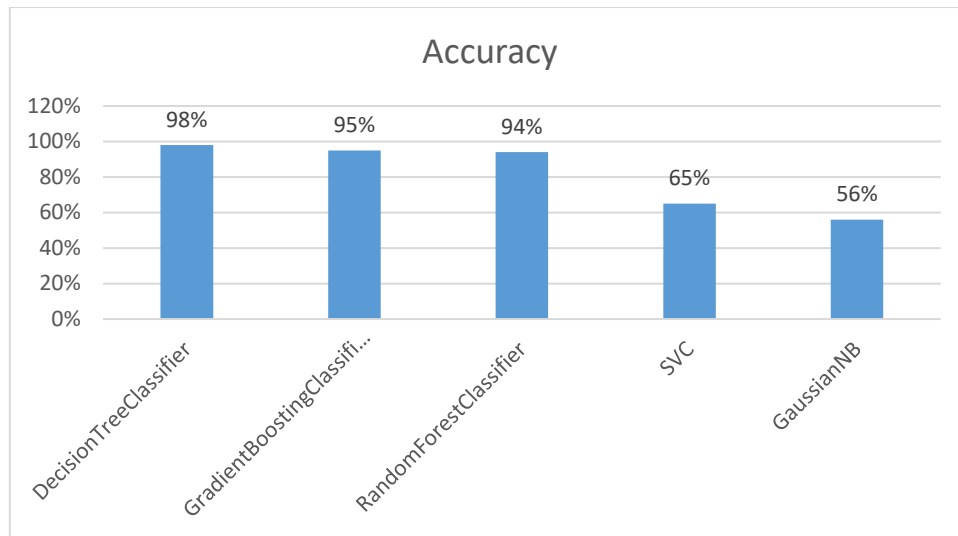


Figure No. 10.2 Comparison of Accuracies of Text Blob Polarity Model

Decision Tree Classifier	Precision	Recall	F1-Score	Support
Negative	.98	.97	.97	828
Positive	.98	.96	.97	458
Accuracy			.98	1286
Macro Average	.98	.96	.97	1286
Weighted Average	.98	.97	.97	1286

Table No 10.2 Decision Tree Classifier – Result.

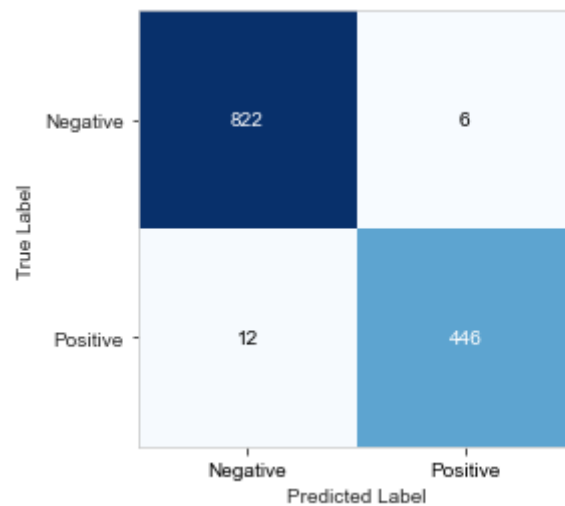


Figure No. 10.3 Decision Tree Confusion Matrix – Text Blob.

KNN-based collaborative filtering model relies on the similarity of item features data distribution without making any assumptions. Distance is calculated between the target/label and every other item label/ target within the dataset. Top k items are returned based on ranks calculated between the distances as shown in Figure No. 10.4 for the credit cards dataset.

```
Recommendations for HDFC Bank Credit Card - fuel
1 : HDFC Bank Credit Card - lifestyle, with a distance of 0.86552552474476
2 : HDFC Bank Credit Card - general, with a distance of 0.8730984066413561
3 : HDFC Bank Credit Card - reward, with a distance of 0.8753430253253338
4 : ICICI Bank Credit Card - fuel, with a distance of 1.0
```

Figure No. 10.4 KNN Collaborative Filtering Models Recommendations.

A chatbot is software that can communicate and perform actions like humans. Retrieval and generative are the basic types of chatbot models based on how they are built. The retrieval-based chatbot uses predefined input patterns and responses, then uses a heuristic approach to select the appropriate response. The loss and accuracy of the model are as shown in Figure No. 10.5.

```
print("model created")
super(SDB, self).__init__(name, kwargs)

Epoch 1/100
7/7 [=====] - 1s 2ms/step - loss: 2.5529 - accuracy: 0.0645
Epoch 2/100
7/7 [=====] - 0s 4ms/step - loss: 2.4569 - accuracy: 0.1290
Epoch 3/100
7/7 [=====] - 0s 2ms/step - loss: 2.3815 - accuracy: 0.1613
Epoch 4/100
7/7 [=====] - 0s 3ms/step - loss: 2.3450 - accuracy: 0.1290
Epoch 5/100
7/7 [=====] - 0s 6ms/step - loss: 2.2048 - accuracy: 0.2581
Epoch 6/100

Epoch 98/100
7/7 [=====] - 0s 2ms/step - loss: 0.2151 - accuracy: 0.9355
Epoch 99/100
7/7 [=====] - 0s 2ms/step - loss: 0.2090 - accuracy: 0.9355
Epoch 100/100
7/7 [=====] - 0s 2ms/step - loss: 0.1328 - accuracy: 1.0000
model created
```

Figure No. 10.5 Loss and Accuracy of the Retrieval Model.

Chapter 11: Deployment

The data pipeline currently consumes the data in file formats to develop the tableau dashboard for sentiment analysis, and credit card recommendations using the collaborative model's data pipeline as shown in Figure No. 11.1 and to develop a retrieval chatbots deployment models diagram as shown in Figure 11.2.

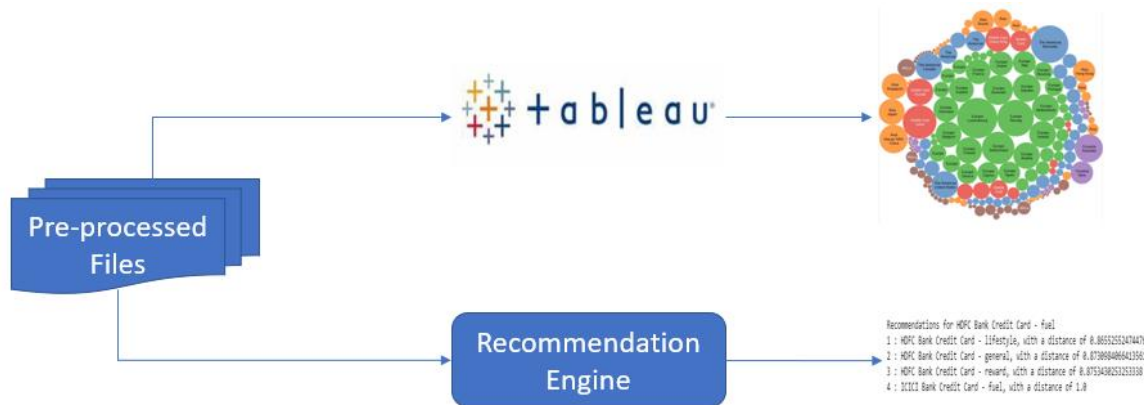


Figure No. 11.1 Deployment Model for tableau dashboard and recommendation engine.

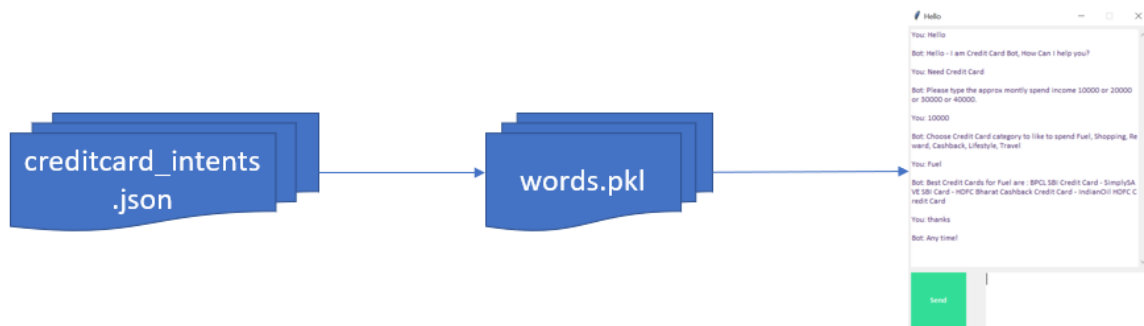


Figure No. 11.2 Deployment Model for Retrieval Chat Bot.

Chapter 12: Analysis and Results

As mentioned in the Data Evaluation section, we got an accuracy of 98% with the Decision Tree Classifier for Text Blob Polarity. There could be various factors affecting the performance of the other classifiers, deep learning, and transfer learning with fine tune BERT models. The model performance can be improved with the following, fine-tuning process:

- Having more datasets.
- Topic modeling with more data points.
- Handling class balance for the bigger dataset.
- Domain-specific support for accurate labeling by specific corpus.
- More efficient negation handling.

SWOT Analysis as shown in Figure No. 12.1.

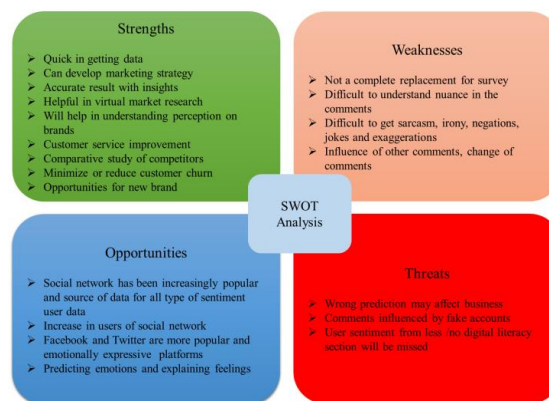


Figure No. 12.1 SWOT Analysis

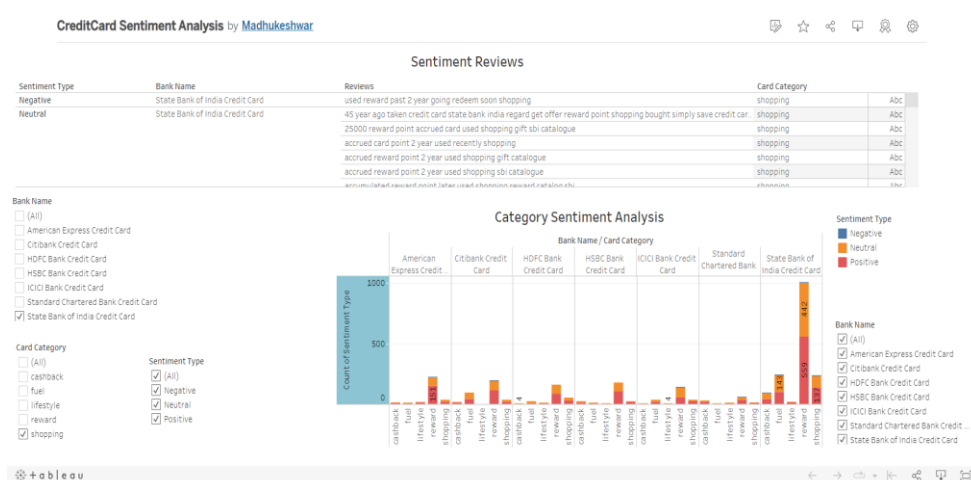


Figure No. 12.2 Credit Card Sentiment Analysis Dashboard

Data extracted for multiple banks and mainly 5 card categories like cashback, fuel, lifestyle, reward, and shopping are from the website. Some of the insights from the dashboard Figure No. 12.2 are:

- SBI credit card transactions are high and Standard Chartered Bank is Low.
- Credit cards are mainly used for remedying the rewards offered by Banks.
- Cards used for the Lifestyle and Cashback category minimal.
- Cards used for fuel usage are moderate.

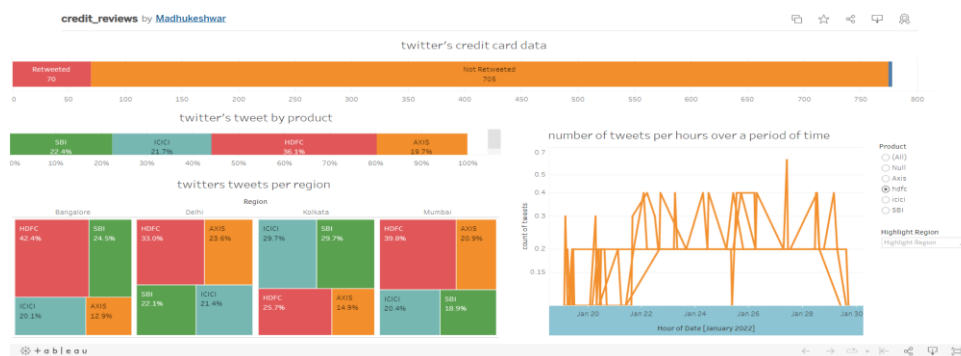


Figure No. 12.3 Credit Card Dashboard by Regions in India

Data extracted for 4 major banks and 4 major cities Bangalore, Delhi, Kolkata, and Mumbai are from Twitter. Some insights from the Figure No. 12.3 dashboard are.

- HDFC Credit Cards are used more for transactions followed by SBI Credit Cards.
- SBI Credit Cards are used more in the region Bangalore and Kolkata.
- ICICI Credit Cards are used in the Kolkata region followed by SBI Credit Cards.
- Axis Credit Cards are used more in Delhi and Mumbai Region.

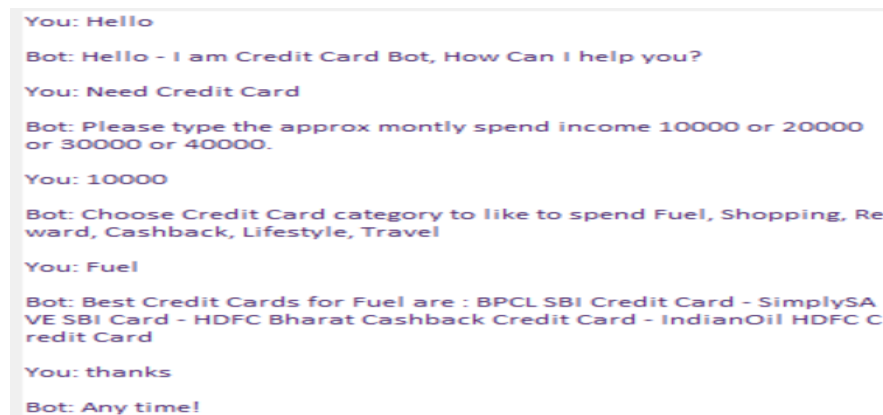
KNN Collaborative Filtering Recommendation framework is focused on the premise that it is possible to use items that are close to an item to speculate how far distance from another item as shown in Figure No. 12.4 for the credit card dataset.

```
Recommendations for State Bank of India Credit Card - general
1 : American Express Credit Card - shopping, with a distance of 0.8786329908905898
2 : American Express Credit Card - reward, with a distance of 0.9510487013222725
3 : American Express Credit Card - general, with a distance of 0.978669514978249
4 : ICICI Bank Credit Card - general, with a distance of 1.0
```

Figure No. 12.4 Recommendation for general credit card

The model was able to predict other bank cards by different categories when one bank card's category was provided with the parameter distance.

Dataset will be used to train a chatbot that contains patterns, responses, and intents using deep learning techniques. RNN is used to classify the user's messages into categories and then from the list of responses, it responds randomly as shown in Figure No. 12.5.

A screenshot of a chatbot interface showing a conversation. The user messages are in blue and the bot responses are in purple. The conversation starts with a greeting, followed by the user asking for a credit card. The bot asks for the user's monthly spend, then for a category of spending. The user selects 'Fuel', and the bot lists several credit cards suitable for fuel spending. The conversation ends with a thank you and a closing message from the bot.

You: Hello

Bot: Hello - I am Credit Card Bot, How Can I help you?

You: Need Credit Card

Bot: Please type the approx montly spend income 10000 or 20000 or 30000 or 40000.

You: 10000

Bot: Choose Credit Card category to like to spend Fuel, Shopping, Reward, Cashback, Lifestyle, Travel

You: Fuel

Bot: Best Credit Cards for Fuel are : BPCL SBI Credit Card - SimplySAVE SBI Card - HDFC Bharat Cashback Credit Card - IndianOil HDFC Credit Card

You: thanks

Bot: Any time!

Figure No. 12.5 Credit card Chat Bot Responses.

Chapter 13: Conclusions and Recommendations for future work

This report highlighted the processes of data wrangling, EDA, and sentiment analysis using various machine learning techniques on data collected from websites and Twitter. A recommendation model was developed with data and finally, a simple QA bot was developed. Currently, the product is not completed automated.

Further work recommendation is to make use of Azure cognitive services and to develop end and end more sophisticated products with the lasted enhanced features.

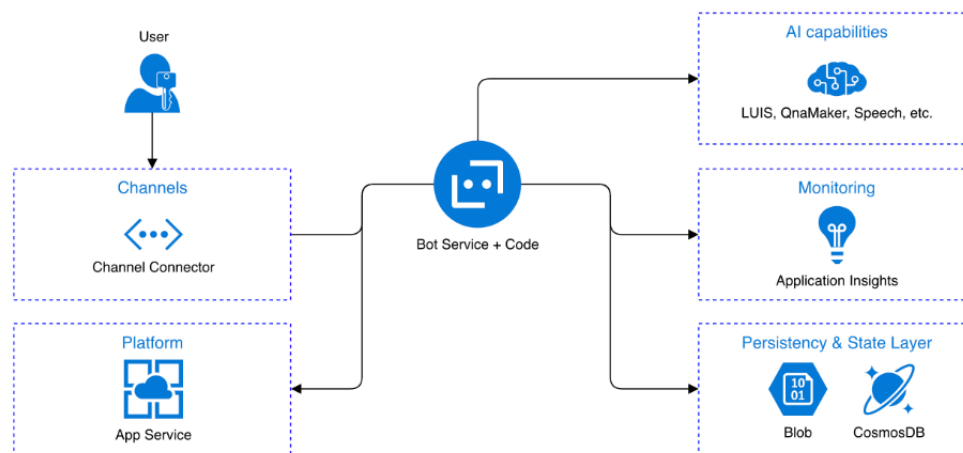


Figure No. 13.1 Bot Architecture on Azure(*Microsoft Bot Framework v4 Explained*) - *Clemens Siebler's Blog*).

APIs get exposed for receiving incoming text and for answering them from centrally placed simple webservises (bot services). Azure App Services holds the Bot Framework which is embedded with an engine for executing the Bot Code. Azure Blob and Cosmos DB are supported as persistency layers for a bot. Bot Code leverages Azure Cognitive Services, which mainly uses Language Understanding Services (LUIS) for NLP as well as for simple question and answer pairs (QNA). Bot Framework also integrates with Application Insights which supports monitoring the bot and its infrastructure. (*Microsoft Bot Framework v4 Explained (JavaScript)* - *Clemens Siebler's Blog*)

Bibliography

- Ahmad, M., Aftab, S., Muhammad, S. S., & Ahmad, S. (2020). Machine Learning Techniques for Sentiment Analysis: A Review. *A Journal of Physical Sciences, Engineering and Technology*, 12(02), 72–78. www.ijmse.org
- Asghar, M. Z., Khan, A., Ahmad, S., Qasim, M., & Khan, I. A. (2017). Lexicon-enhanced sentiment analysis framework using rule-based classification scheme. *PLoS ONE*, 12(2), 1–22. <https://doi.org/10.1371/journal.pone.0171649>
- Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2014). *Research Paper Recommender Systems : A Literature Survey Table of Content*. 1–68. [https://www.scss.tcd.ie/joeran.beel/pubs/2016 IJDL --- Research Paper Recommender Systems -- A Literature Survey \(preprint\).pdf](https://www.scss.tcd.ie/joeran.beel/pubs/2016 IJDL --- Research Paper Recommender Systems -- A Literature Survey (preprint).pdf)
- Breaking through text clutter with natural language processing (NLP)*. (n.d.). Retrieved June 27, 2022, from <https://www.latentview.com/blog/breaking-text-clutter-natural-language-processing/>
- Chakrabarti, S., Trehan, D., & Makhija, M. (2018). Assessment of service quality using text mining – evidence from private sector banks in India. *International Journal of Bank Marketing*, 36(4), 594–615. <https://doi.org/10.1108/IJBM-04-2017-0070>
- Credit Card Industry Analysis - Overview, Market Dynamics, Costs*. (2021, September 5). <https://corporatefinanceinstitute.com/resources/knowledge/credit/credit-card-industry-analysis/>
- Credit card spends hit Rs 2 lakh crore high in Q2 - Times of India*. (n.d.). Retrieved July 12, 2022, from <https://timesofindia.indiatimes.com/business/india-business/credit-card-spends-hit-rs-2-lakh-crore-high-in-q2/articleshow/86820265.cms>
- Cross-industry standard process for data mining - Wikipedia*. (2022, April 5). https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining#cite_note-Harper06-14
- Haller, E., & Rebedea, T. (2013). Designing a chat-bot that simulates an historical figure. *Proceedings - 19th International Conference on Control Systems and Computer Science, CSCS 2013*, 582–589. <https://doi.org/10.1109/CSCS.2013.85>
- History of the Credit Card*. (2017). https://en.wikipedia.org/wiki/Credit_card
- Javkar, K. G., Vora, S. H., Rodge, A. S., Bose, J., & Sharma, H. (2016). Best offer recommendation service. *2016 International Conference on Advances in Computing*,

- Communications and Informatics, ICACCI 2016*, 2430–2436.
<https://doi.org/10.1109/ICACCI.2016.7732421>
- Li, L., & Wang, L. (2020). News recommendation based on content fusion of user behavior. *Proceedings - 2020 13th International Symposium on Computational Intelligence and Design, ISCID 2020, 1*, 217–220. <https://doi.org/10.1109/ISCID51228.2020.00055>
- Microsoft Bot Framework v4 explained (JavaScript) - Clemens Siebler's Blog*. (n.d.). Retrieved July 11, 2022, from <https://clemenssiebler.com/microsoft-bot-framework-v4-explained-javascript/>
- Mittal, D., & Agrawal, S. R. (2022). Determining banking service attributes from online reviews: text mining and sentiment analysis. *International Journal of Bank Marketing*, 40(3), 558–577. <https://doi.org/10.1108/IJBM-08-2021-0380>
- P, R. K. M. E. A. (2020). *Sentiment analysis in E-Commerce using Recommendation System*. 8(12), 114–119.
- Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web, WWW 2001*, 285–295. <https://doi.org/10.1145/371920.372071>
- Sentiment Analysis in banking - Maveric Systems*. (n.d.). Retrieved August 10, 2022, from <https://maveric-systems.com/blog/sentiment-analysis-in-banking/>
- Shah, A. (2021). Sentiment analysis of product reviews using supervised learning. *Reliability: Theory and Applications*, 16, 243–253.
<https://doi.org/10.1145/3447568.3448513>
- Shaikh, S., Rathi, S., & Janrao, P. (2017). Graph Based Approached. *2017 IEEE 7th International Advance Computing Conference*, 932–935.
<https://doi.org/10.1109/IACC.2017.180>
- Umuhoza, E., Ntirushwamaboko, D., Awuah, J., & Birir, B. (2020). Using unsupervised machine learning techniques for behavioral-based credit card users segmentation in Africa. *SAIEE Africa Research Journal*, 111(3), 95–101.
<https://doi.org/10.23919/saiee.2020.9142602>
- Xue, H., & Zhang, D. (2019). *Based on Content and Social Network. Itaic*, 477–481.
- Zaza, S., & Al-Emran, M. (2016). Mining and exploration of credit cards data in UAE. *Proceedings - 2015 5th International Conference on e-Learning, ECONF 2015*, 275–279. <https://doi.org/10.1109/ECONF.2015.57>

Appendix

Plagiarism Report¹

Sentiment Analysis on Credit Cards using Online Reviews

by Madhukeshwar K

Submission date: 29-Jul-2022 06:22PM (UTC+0530)

Submission ID: 1876549960

File name: Sentiment_Analysis_on_Credit_Cards_using_Online_Reviews.docx (2.6M)

Word count: 5295

Character count: 29968

¹ Turnitn report to be attached from the University.

Sentiment Analysis on Credit Cards using Online Reviews

ORIGINALITY REPORT

10%	9%	3%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Queen Mary and Westfield College Student Paper	3%
2	corporatefinanceinstitute.com Internet Source	2%
3	www.coursehero.com Internet Source	1%
4	Submitted to Lovely Professional University Student Paper	1%
5	www.analyticsvidhya.com Internet Source	1%
6	Submitted to Monash University Student Paper	1%
7	en.wikipedia.org Internet Source	<1%
8	www.amrita.edu Internet Source	<1%
9	Submitted to Universiti Malaysia Perlis Student Paper	<1%

10	Submitted to University of Portsmouth Student Paper	<1 %
11	etd.uum.edu.my Internet Source	<1 %
12	Kalyan Sahu, Yu Bai, Yoonsuk Choi. "Supervised Sentiment Analysis of Twitter Handle of President Trump with Data Visualization Technique", 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020 Publication	<1 %
13	Submitted to University of Bridgeport Student Paper	<1 %
14	www.wishfin.com Internet Source	<1 %

Exclude quotes On
Exclude bibliography On

Exclude matches < 10 words

Publication in a Conference Presented

Madhukeshwar R K “A Recommender System for Indian Credit Cards using Text Analytics.”
International Conference on Machine Learning & Data Science Innovations (MacDat'22).



Publication in a Journal

Madhukeshwar R K, Ratnakar Pandey, Shinu Abhi, "A Recommender System for Indian Credit Cards using Text Analytics." NeuroQuantology, Volume 20, Issue8, 2022, Page 9021-9028, doi:10.14704/nq.2022.20.8.NQ44922

NeuroQuantology | July 2022 | Volume 20 | Issue 8 | Page 9021-9028 | doi:10.14704/nq.2022.20.8.NQ44922
Madhukeshwar R K et al/ A Recommender System for Indian Credit Cards using Text Analytics



A Recommender System for Indian Credit Cards using Text Analytics

Madhukeshwar R K

madhukeshwar.ba06@reva.edu.in

REVA Academy for Corporate Excellence, REVA University, Bengaluru, India

Ratnakar Pandey

ratnakarpandey@race.reva.edu.in

REVA Academy for Corporate Excellence, REVA University, Bengaluru, India

Shinu Abhi

shinuabhi@reva.edu.in

REVA Academy for Corporate Excellence, REVA University, Bengaluru, India

Abstract

With the increase in web and social media usage, views and experiences on products or services shared by online users have increased. Social media acts as the main source of data for text analytics on which the users' sentiments can be performed. Organizations can gain valuable insights into social marketing strategies by finding sentiments and emotions towards their products. Sentiment analysis is a process of computationally finding, classifying, and categorizing opinions expressed on the block of text, to decide whether sentiment towards a particular topic or a product, is positive, negative, or neutral and whether emotions are happy, sad, angry, etc. by combining Machine Learning and Natural Language Processing. In this paper, we perform sentiment analysis through NLP techniques on the reviews and tweets collected from websites and Twitter on prominent Indian credit cards. Predicted sentiment values are used to develop a recommendation model which recommends similar credit cards based on categories.

9021

Keywords: Text Mining, Sentiment Analysis, Sentiment Classification, Natural Language Processing, Recommendation engine, Credit Cards.

DOI Number: 10.14704/nq.2022.20.8.NQ44922

NeuroQuantology 2022; 20(8): 9021-9028

1. INTRODUCTION

Credit cardholders make credit payments to buy products or to the required services using a payment card which is commonly known as a credit card. Usually, this is associated with the revolving account that gives credit to the cardholder to make payment for his / her needs and later pay back to the card issuer within the agreed time limit. Credit cards can be classified into two groups i.e. consumer cards and business cards (O'Sullivan et al., 2003). On high-value purchases, the cardholder can convert the total amount of purchases into low-cost EMI to enable easy repayment over a period thus revolutionizing their shopping experience.

Banks issue credit cards with a credit limit,

eISSN 1303-5150

allowing the cardholder to make payments. The card limit is based on the cardholder's income, credit score, and bank account transaction history. Repayment of spent amount using the credit card need can be done without paying interest, by making the repayment within the predefined period (Credit Card Industry Analysis - Overview, Market Dynamics, Costs, 2021). Learning how to use a credit card and how to make use of the credit card period and repay the amount on time helps cardholders to boost their credit score and help them to get better eligibility for high credit without any difficulty.

In the last few years, custom-designed credit cards based on the type of usage have become a big selling point. Co-branded and

www.neuroquantology.com



affinity cards have become more popular than ever (*History of the Credit Card*, 2017). Credit cards custom designed for travel at the airport or railway station gives the cardholders a unique experience through complimentary lounge access, priority check-in, discounts at the restaurants and more. A few of the travel cards also cover comprehensive travel insurance. Discounts on credit

1

cards get extended on movie tickets, online shopping, health, and wellness outlets, and surcharge waivers at petrol/diesel pumps across the country.

Credit card customers use social media and online review portals to post their experiences and opinions openly with the world. The availability of this huge customer data has pulled business users and researchers from different fields to better understand their customer sentiments. Sentiment analysis' primary objective is to predict the polarity of the text i.e. its positive, negative, or neutral (*Sentiment Analysis & Machine Learning Techniques - Data Analytics*, 2021).

This paper intends to design a recommendation system to help the users to select the right card for the right occasion based on the collected users reviews from websites and predict their sentiment polarity using machine learning techniques (*Sentiment Analysis - Wikipedia*).

2. LITERATURE REVIEW

This section examines the extant literature available on the need for developing a recommendation system based on sentiment analysis with natural language techniques in the credit card market.

Given the fierce competition in the global market the credit card industry is highly vulnerable to customer power in the emerging e-payment and fintech industry. Innovation in financial services to provide a personalized experience to customers with a better offer on credit card transactions become inevitable (Umuhoza et al., 2020). Mining credit card reviews can help the banks to find interesting patterns among different variables that may be used in the future to design better products (Zaza & Al Emran, 2016).

The digital transformation in the banking
 eISSN1303-5150

sector has enforced e-payment customers to purchase products online by staying at home and due to the proliferation of online stores, online reviews have also increased as the source of information on product quality and durability. Opinion mining on these online reviews helps customers to make a better decision on the purchase (Mittal & Agrawal, 2022).

Online shopping has become increasingly popular due to the variety of products, lower prices, availability of different models/brands, and fast logistic systems. The explosion of offers has forced the shoppers to make use of debit/credit cards and to avail of this cashback and discounts offered by card issuers. Credit card holders typically aggregate reward points through various offers from multiple credit cards. Cashback or rewards acquired from the credit card transactions varies in percentage from the issuer of the credit cards, identifying the best reward or cashback for a given card is difficult (Javkar et al., 2016).

Post shopping, shoppers provide their ratings, review, and emotions on websites which becomes the main source for purchaser's sentiments data generation. Multiple tools and techniques are available in the market for automatically classifying the sentiments for user-generated data. Sentiment analysis helps users to make better purchases through their collective analysis of sentiments. In deep learning models, the network learns to extract the features while the learning/training process. Word2vec modeling technique uses CNN to get trained and to classify the sentiments on reviews collected (Shah, 2021).

User text data can also be fed to a stochastic learning algorithm that analyses and classifies the feedback as negative, positive, and neutral and provides recommendations to shoppers for their next purchases (P, 2020). Lexicon, machine learning, or a hybrid combination of both are the most commonly used approach (Ahmad et al., 2020).

2

Lexicon Based algorithms can classify the user sentiment through polarity score or using a machine learning classifier to identify specific text into a sentiment class. Two problems to be solved here are subjectivity classification; a text is subjective or objective and polarity

www.neuroquantology.com

9022



classification; the text is a positive or negative or neutral (*Sentiment Analysis in Banking - Maveric Systems*, n.d.). The lexicon based approach is easy to understand and implement (Chakrabarti et al., 2018). However, user-shared reviews raise challenges due to insufficient coverage of emotions expressed. With an unsupervised approach, accuracy is determined by the classifier that might need modifications or negations (Asghar et al., 2017).

Recommendation systems typically are classified into content, collaborative, and hybrid-based recommendations. When properties of targets are considered for the recommendation it is called content based. When the system recommends the targets based on the comparison measures between other targets and users it's called collaborative filtering. A hybrid recommendation is based on a combination of content based and collaborative (Shaikh et al., 2017).

Content-based algorithms come with limitations of lack of diversified reviewer's interests, so the content fusion of reviewer behavior is suggested. It is implemented by building the correlation between the popularity of the reviewer's interest and the text and then finding the user preferences along with time utility and finally fusing the potential and user preferences to provide a recommendation list (Li & Wang, 2020).

In the collaborative filtering technique, the number of users increases the amount of work required by the system. The technique should be able to provide quality recommendations for complex problems. For complex problems,

the preferred technique is item-based collaborative filtering. The item-based technique uses indirect computing recommendations for the user from the relationship identified between different targets which is an output of the user-target matrix (Sarwar et al., 2001).

Xue and Zhang propose to calculate a new distance between the short and long text's similarity as a technique to identify the nearest neighbor set from the social network of the user and recommend the texts to the user's nearest neighbor set (Xue & Zhang, 2019).

A study on common recommendation techniques reveals that 55% of approaches are content based filtering, around 18% are collaborative filtering, and 16% are graph-based recommendations. Hybrid recommendations, stereotyping and item-centric recommendations are the other techniques that are applied (Beel et al., 2014).

3. METHODOLOGY

Natural Language Processing (NLP) falls under the branch of Artificial Intelligence (AI) which is a branch of Computer Science. It mainly provides an understanding ability of computers like the way human beings can text and speak words. NLP includes rule-based modeling, computational linguistics with Machine Learning (ML), Statistics, and Deep Learning Models as shown in Fig 3.1. With a combination of these technologies, text data produced out of human languages is understood by the computers to provide meaning to the full user's sentiments and intents.

3

9023



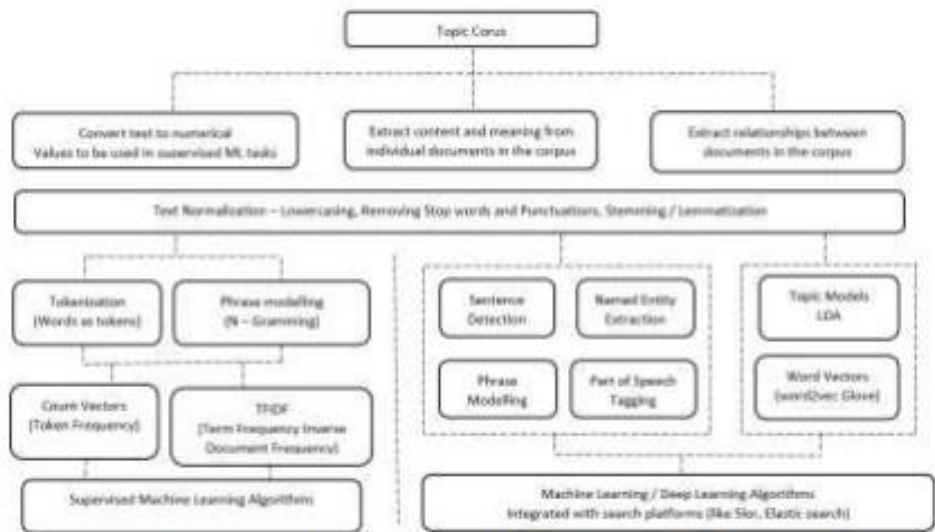


Figure 3.1 NLP Flow Diagram (Breaking through Text Clutter with Natural Language Processing (NLP))

The data for this study, user reviews from the website, and tweets are collected using a trial version of the web scraping tool – Octoparse and using scraps and tweets from the Twitter API using R programming. Various text preprocessing activities were done on the text. We have used the lexicon method for scoring. Exploratory data analysis is done on the frequency of words and repeated words, followed by feature extraction. Here we transform the unstructured text data into machine-readable format and numbers using a bag of words, TF-IDF, and word embedding. We did sampling and used various classifiers and topic detection models to provide the

sentiment of texts as positive, negative, and neutral on the reviews and tweets.

4. PROPOSED SYSTEM

Data collected from the websites and tweets are preprocessed as mentioned in the data pipeline diagram (Figure 4.1). Each text undergoes text normalization i.e., converting the text to lowercase, removing the stop words, punctuations, stemming, and lemmatization. EDA is performed on the texts for word frequency and word cloud followed by feature extraction. Finally, a separate set of datasets are created and stored in the datastore. This data is further processed to achieve the required dataset.

9024



Figure 4.1 Data pipeline for data preprocessing

Dataset from the data store is further processed for building a recommendation engine pipeline (Figure 4.2). Data in the dataset is filtered which had only positive sentiments. Two features credit card and card category are combined to get a unique feature called *credit card category*. The final dataset is reduced to needed features followed by binning the polarity.

4



Figure 4.2 Data Preparation Steps for Recommendation Engine

5. MODELING

Preprocessed data was fed into multiple

models to get the polarity value for the sentiments. The classification technique used

etSSN1303-5150

www.neuroquantology.com



was the Lexicon-Based approach, Text Blob Sentiment Analysis, VADER sentiment analysis, TF-IDF Vectorizer, and Fine Tuning with BERT.

Text Blob is a python package that calculates sentiment scoring based on the polarity of the dataset from -1 to 1 (Hermansyah & Sarno, 2020). It consists of a large number of corpora sets and provides stemmers and algorithms to perform text analysis. KNN is a regression and classification machine learning technique, it looks at the labels of several data points near

a targeted data point to make an educated guess regarding the data category. Even though it is straightforward, KNN is a powerful machine learning technique (Gafoor et al., 2022).

6. ANALYSIS AND RESULTS

Accuracy of Lexicon with AFFIN vocabulary, classifiers with Text Blob Polarity, VADER Sentiment, TF-IDF Vectorizer, and Transfer Learning using fine-tuned BERT Model are compared. It is important to have accurate labeling for all comments.

9025

S.N.	Approach	Accuracy	Classifier with Best Result
1.	Text Blob Polarity	98%	Decision Tree Classifier
2.	Fine Tuning with BERT	97%	Transfer Learning
3.	Word Embedding TF-IDF Vectorizer	94%	Gradient Boosting Classifier
4.	Lexicon Vocabulary	93%	Gradient Boosting Classifier
5.	VADER Sentiment	90%	Random Forest Classifier

Table 5.1 Modeling Accuracy

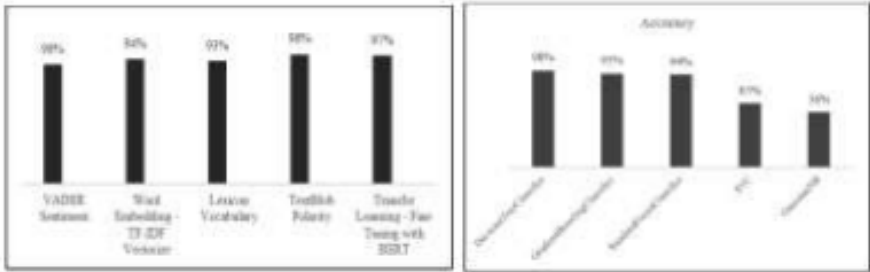


Figure 5.1 Comparison of model accuracies Figure 5.2 Comparison of accuracies Text Blob Polarity Model

Figure (5.1) shows that the classifier Text Blob Polarity classifier has the highest accuracy of 98% for the sentiment analysis. VADER Sentiment classifier with the lowest accuracy of 90%. Text Blob Polarity with Decision Tree Classifier has given the highest accuracy of

98% for the sentiment analysis.
5
There are numerous factors affecting the model performance of other approaches, especially for deep learning techniques for which we need to have more optimized data.

Decision Tree Classifier	Precision	Recall	F1-Score	Support
Negative	.98	.97	.97	828
Positive	.98	.96	.97	458
Accuracy			.98	1286



Macro Average	.98	.96	.97	1286
Weighted Average	.98	.97	.97	1286

Table 5.2 Decision Tree Classifier – Result.

Table 5.2 shows that the Decision Tree classifier for Text Blob Polarity has an accuracy of 98%, precision and recall for the positive classifier being 98% and 96% and F1-Score being 0.97.

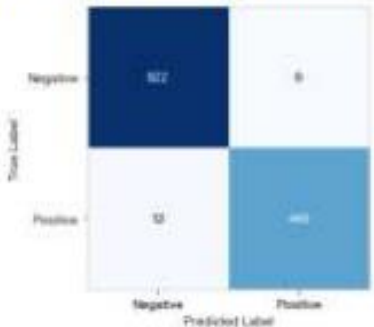


Figure 5.3 Decision Tree Confusion Matrix – Text Blob.

9026

KNN-based collaborative filtering model relies on the similarity of item features and data distribution without making any assumptions. Distance is calculated between the target/label and every other item label/ target within the dataset. Top items are returned based on ranks calculated between the distances.

Recommendations for HDFC Bank Credit Card - fuel
1 : HDFC Bank Credit Card - lifestyle, with a distance of 0.86352552474476
2 : HDFC Bank Credit Card - general, with a distance of 0.8730984066413561
3 : HDFC Bank Credit Card - reward, with a distance of 0.8753430253253338
4 : ICICI Bank Credit Card - fuel, with a distance of 1.0

Figure 5.4 KNN Collaborative Filtering Models Recommendations.

The Decision Tree Classifier for Text Blob Polarity has 98% accuracy. There could be several factors affecting the performance of the other classifiers like deep learning and transfer learning with fine-tuned BERT models. The model performance can be improved with the following fine-tuning process.

- Having more datasets.
- Topic modeling with more data points.
- Handling class balance for the bigger dataset.
- Domain-specific support for accurate labeling by specific corpus.
- More efficient negation handling.

6

KNN Collaborative Filtering Recommendation framework is focused on the premise that it is possible to use items that are close to the item to guess how far distance from another item is shown in Figure 5.7.

Recommendations for State Bank of India Credit Card - general
1 : American Express Credit Card - shopping, with a distance of 0.8780329900000509
2 : American Express Credit Card - reward, with a distance of 0.951048701322725
3 : American Express Credit Card - general, with a distance of 0.978660514076340
4 : ICICI Bank Credit Card - general, with a distance of 1.0

Figure 5.7 Recommendation for general credit card

The model was able to predict other bank cards by distinct categories when one bank card's category was provided with the parameter distance.

7. CONCLUSION

The purpose of this paper is to perform text

analysis and sentiment analysis on the reviews and tweets collected from websites and Twitter and to develop a tableau-based dashboard and a credit card recommendation engine for the users, to choose the right credit card for the right offer or discounts offered by



the credit card issuer.

This study uses sentiment analysis using various text analytics approaches to find whether a text is negative or positive. Significant approaches were used like Lexicon with AFFIN vocabulary, classifiers with Text Blob Polarity, VADER Sentiment, TF-IDF Vectorizer, and Transfer Learning using fine tuned BERT Models. Text Blob Polarity with Decision Tree Classifier has given the highest accuracy and hence was used for creating the polarity values. Polarity values were used to feed the KNN Collaborative Filtering Model to recommend the credit cards based on the category to recommend the other credit cards category. Retrieval Credit Card was developed using Deep Learning.

8. REFERENCE

Ahmad, M., Aftab, S., Muhammad, S. S., & Ahmad, S. (2020). Machine Learning Techniques for Sentiment Analysis: A Review. *A Journal of Physical Sciences, Engineering and Technology*, 12(02), 72–78. www.ijmse.org

Asghar, M. Z., Khan, A., Ahmad, S., Qasim, M., & Khan, I. A. (2017). Lexicon-enhanced sentiment analysis framework using rule-based classification scheme. *PLoS ONE*, 12(2), 1–22.
<https://doi.org/10.1371/journal.pone.0171649>

Beel, J., Gipp, B., Langer, S., & Breitinger, C. (2014). *Research Paper Recommender Systems: A Literature Survey Table of Content*. 1–68.
[https://www.scss.tcd.ie/joeran.beel/pubs/2016_IJDL -- - Research Paper Recommender Systems -- A Literature Survey \(preprint\).pdf](https://www.scss.tcd.ie/joeran.beel/pubs/2016_IJDL_-_Research_Paper_Recommender_Systems_-_A_Literature_Survey_(preprint).pdf)

Breaking through text clutter with natural language processing (NLP). (n.d.). Retrieved June 27, 2022, from <https://www.latentview.com/blog/breaking-text-clutter-natural-language-processing/>

Chakrabarti, S., Trehan, D., & Makhija, M. (2018). Assessment of service quality using text mining – evidence from private sector banks in India. *International Journal of Bank Marketing*, 36(4), 594–615.
<https://doi.org/10.1108/IJBM-04-2017-0070>

Credit Card Industry Analysis - Overview, Market Dynamics, Costs. (2021, September 5). <https://corporatefinanceinstitute.com/resources/knowledge/credit/credit-card-industry-analysis/>

eISSN1303-5150

Gafoor, A., Srujana, A. L., Nagasri, A., Durgaprasad, G. S. S., & Dasari, L. S. K. (2022). KNN based Entertainment Enhancing System. *2022 6th International Conference on Trends in Electronics*

7
and Informatics, ICOEI 2022 - Proceedings, Icoei, 1056–1061.

<https://doi.org/10.1109/ICOEI53556.2022.9777225>

Hermansyah, R., & Sarno, R. (2020). Sentiment Analysis about Product and Service Evaluation of PT Telekomunikasi Indonesia Tbk from Tweets Using TextBlob, Naive Bayes & K-NN Method. *International Sem Inar on Application for Technology of Information and Communication*, 511– 516.

History of the Credit Card. (2017). https://en.wikipedia.org/wiki/Credit_card

Javkar, K. G., Vora, S. H., Rodge, A. S., Bose, J., & Sharma, H. (2016). Best offer recommendation service. *2016 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2016*, 2430–2436.
<https://doi.org/10.1109/ICACCI.2016.7732421>

Li, L., & Wang, L. (2020). News recommendation based on content fusion of user behavior. *Proceedings - 2020 13th International Symposium on Computational Intelligence and Design, ISCID 2020*, 1, 217–220.
<https://doi.org/10.1109/ISCID51228.2020.00055>

Mittal, D., & Agrawal, S. R. (2022). Determining banking service attributes from online reviews: text mining and sentiment analysis. *International Journal of Bank Marketing*, 40(3), 558–577.
<https://doi.org/10.1108/IJBM-08-2021-0380>

O'Sullivan, A., Sheffrin, S., & Perez, S. (2003). *Microeconomics: Principles, Applications and Tools, Student Value Edition (9th Edition)*. 29.

P, R. K. M. E. A. (2020). *Sentiment analysis in E-Commerce using Recommendation System*. 8(12), 114–119.

Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. *Proceedings of the 10th International Conference on World Wide Web, WWW 2001*, 285–295.
<https://doi.org/10.1145/371920.372071>

Sentiment analysis - Wikipedia. (n.d.). www.neuroquantology.com



9027

Retrieved July 24, 2022, from
https://en.wikipedia.org/wiki/Sentiment_analysis#cite_note-SentiStrength2010-17
Sentiment Analysis & Machine Learning Techniques - Data Analytics. (2021).
<https://vitalflux.com/sentiment-analysis-machine-learning-techniques/>
Sentiment Analysis in banking - Maveric Systems. (n.d.). Retrieved August 10, 2022, from
<https://maveric-systems.com/blog/sentiment-analysis-in-banking/>
Shah, A. (2021). Sentiment analysis of product reviews using supervised learning. *Reliability: Theory and Applications*, 16, 243–253.
<https://doi.org/10.1145/3447568.3448513>
Shaikh, S., Rathi, S., & Janrao, P. (2017). Graph Based Approached. *2017 IEEE 7th International Advance Computing Conference*, 932–935.
<https://doi.org/10.1109/IACC.2017.180>
Umuhoza, E., Ntirushwamaboko, D., Awuah, J., & Birir, B. (2020). Using unsupervised machine learning techniques for behavioral-based credit card users segmentation in Africa. *SAIEE Africa Research Journal*, 111(3), 95–101.
<https://doi.org/10.23919/saiee.2020.9142602>
Xue, H., & Zhang, D. (2019). *Based on Content and Social Network. Itaic*, 477–481.
Zaza, S., & Al-Emran, M. (2016). Mining and exploration of credit cards data in UAE. *Proceedings - 2015 5th International Conference on e-Learning, ECONF 2015*, 275–279.
<https://doi.org/10.1109/ECONF.2015.57>

8

9028



Github Link

<https://github.com/kyasanur/Sentiment-Analysis-on-Credit-Cards-using-Online-Reviews>