RV Educational Institutions ®
RV College of Engineering ®

Autonomous Institution Affiliated to Visvesvaraya Technological University, Belagavi

Approved by AICTE, New Delhi, Accredited By NAAC, Bengaluru And NBA, New Delhi

Go, change the world

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Ad Analysis Using Machine Learning

PROJECT REPORT

Submitted by,

Name 1 : Yashas C N **USN1 : 1RV16CS184**

Name2 : Kishan V **USN2 : 1RV16CS192**

Name3 : Pranay Reddy Juturu **USN3 : 1RV16CS064**

Under the guidance of

Prof. Pavithra H
Assistant Professor
Dept. of CSE
RV College of Engineering

In partial fulfilment for the award of degree
Of
Bachelor of Engineering
In
Computer Science and Engineering
2019-2020

RV COLLEGE OF ENGINEERING®, BENGALURU-59

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the major project work titled ***'Ad analysis using Machine Learning'*** is carried out by **Yashas C N (1RV16CS1840), Kishan V (1RV16CS192), and Pranay Reddy Juturu (1RV16CS064)** who are bonafide students of RV College of Engineering, Bengaluru, in partial fulfilment for the award of degree of **Bachelor of Engineering in Computer Science and Engineering** of the Visvesvaraya Technological University, Belagavi during the year 2018-2019. It is certified that all corrections/suggestions indicated for the Internal Assessment have been incorporated in the major project report deposited in the departmental library. The major project report has been approved as it satisfies the academic requirements in respect of major project work prescribed by the institution for the said degree.

Signature of Guide	**Signature of Head of the Department**	**Signature of Principal**
Prof. Pavithra H	**Dr. Ramakanth Kumar P**	**Dr.K.N.Subramanya**

External Viva

Name of Examiners	**Signature with Date**
1	
2	

RV COLLEGE OF ENGINEERING®, BENGALURU-59

(Autonomous Institution Affiliated to VTU, Belagavi)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DECLARATION

We, Yashas C N, Kishan V, Pranay Reddy Juturu students of eighth semester B.E., department of CSE, RV College of Engineering, Bengaluru, hereby declare that the major project titled **'Ad Analysis using Machine Learning'** has been carried out by us and submitted in partial fulfilment for the award of degree of **Bachelor of Engineering** in **Computer Science and Engineering during** the year 2019-20.

Further we declare that the content of the dissertation has not been submitted previously by anybody for the award of any degree or diploma to any other university.

We also declare that any Intellectual Property Rights generated out of this project carried out at RVCE will be the property of RV College of Engineering, Bengaluru and we will be one of the authors of the same.

Place: Bengaluru

Date: 02-07-2020

Name **Signature**

1. Yashas C N (1RV16CS184)

2. Kishan V (1RV16CS192)

3. Pranay Reddy Juturu (1RV16CS064)

ACKNOWLEDGEMENT

We are indebted to our guide, **Prof. Pavithra H**, **Assistant Professor**, **Department of CSE** for her wholehearted support, suggestions and invaluable advice throughout our project work and also helped in the preparation of this thesis.

Our sincere thanks to **Dr. Ramakanth Kumar P.**, Professor and Head, Department of Computer Science and Engineering, RVCE for his support and encouragement.

We express sincere gratitude to our beloved Principal, **Dr. K. N. Subramanya** for his appreciation towards this project work.

We thank all the **teaching staff and technical staff** of Computer Science and Engineering department, RVCE for their help.

Lastly, we take this opportunity to thank our **family** members and **friends** who provided all the backup support throughout the project work.

ABSTRACT

Recommendation system is a type of a system in machine learning where the recommendation of a data is given based on the previous data. In Ad Analysis, the ads are recommended to the user based on their priority. In today's generation most of the social media platforms are generating income through ads, so it really important for these organization to target right ads to the right so that, it'll be a win-win situation for both the organization and the users.

In our project, ad analysis using machine learning, we are using Gaussian naïve Bayes, 3 NLP algorithms namely (Bag of words, tokenization, and lemmatization) to cancel the stop words, punctuations and special characters. And we are using sentiment analysis, for characterizing different emotion and classifying. Then we are using stacked ensemble model to combine two different models to get better accuracy and result. So, all of these models and their accuracy and result will be given in detail in the chapters of this document.

From our observation, we have found that, from using these algorithms we have achieved ~94% accuracy. After using stacked ensemble model, we are able to find optimal solution for this project. Regarding the future work of the project, we can extend the application by creating a GUI or by having our datasets in a database or by integrating the application in to a website like Facebook, Instagram etc… So, the all these topics is explained in detail in this document.

ACRONYMS

ANN – ARTIFICIAL NEURAL NETWORKS

CNN – CONVOLUTION NEURAL NETWORKS

AI – ARTIFICIAL INTELLIGENCE

ML-MACHINE LEARNING

TABLE OF CONTENTS

List of Tables

Page No

List of Figures

Page No

CHAPTER-1

INTRODUCTION TO AD ANALYSIS

INTRODUCTION

Machine learning is an area of computer science, statistic theory, and data optimization theory which allows cumbersome tasks to be solved for which a definite manner or the logical approach wouldn't be feasible. In other words, it is a form of training a computer to improve predictions or features based on some input data. The data inputted depends entirely on the problem. It could be data obtained from the machines like computers connected to the same network or the different network, results obtained from the program as an input to the other program.The social media is a computer- conciliate tool that allows people and organizations to formulate, share, create and exchange data, career interests, ideologies, pictures/videos in virtual communities and networks. It allows the development of online social networks by connecting a user's profile with other individuals and/or groups. Since people post unstructured content in social media, it is relatively difficult to extract product related information from such content. Using a users' profile, it is easy such as interests, likes, hobbies, thoughts and so on. However, social media content has a huge potential in delivering more personalized advertisements. Most of the commercial advertisements and web applications are not based on users' actual preferences. They do not have social media related advertisements classification system, self-updating or a continuously updating user character profile more personalization and preference-based advertisements pushing mechanism. Therefore users, advertisers, and corporations are not often harnessing the full potential of social media-based advertising. In this paper, we present an efficient way to provide personalized advertisements to the right people at the right time as well as advertisements pushing mechanism based on user preferences. We make use of ontology mapping and semantic analysis, advertisements classification mechanism to achieve this target in our AdSeeker advertisement engine.

1.1 Motivation

Due to a large section of advertisements being carried out on social media, this project has been chosen with interest to build a more efficient system with better targeted audience by analysing the source of advertisement and behaviour.

1.2 Problem Statement

The key purpose of doing this project is to build a recommendation system by Ad analysis. Customers these days are dependent on recommendations whether it is for products to purchase, news on recent launches, restaurants to visit or services to avail. One other challenge is to Recommend the Ads based on the user types. Hence the root to provide the necessary recommendation is Ads is the Recommender system and also it is important to identify the provided Advertisements are Legitimate. Recommender systems solve this problem of searching through large volume of dynamically generated information to provide users with personalized contents and services. In this project, we attempt to understand different kind of algorithms for Analyzing the Ads systems and compare their performance on E-commerce dataset which is picked from the known e-commerce site. The reason to pick up the dataset from sites is straight simple that there is bucket list of different aspects of our analysis. We have used Supervised Learning Algorithms like Gaussian Naïve and also implemented Natural Language Processing using Bag of Words, TF-IDF and Hashing for Sentiment Analysis, and Stacked Ensemble model for our recommender system.

1.3 Objectives

The overall objective of this thesis project is to investigate the relation between sentiment extracted from social media and the performance of generating Ads. The goal is to analysis the User reviews on a product and specific Ads related to the product and draw a relationship to analysis Ads and recommend.

This results in the following more specific objectives:

- Develop techniques for sentiment analysis of data from Social medias
- Use machine learning and train a model to predict the performance of a site
- Develop a prototype tool for the proposed method

1.3 Scope

This project is done keeping in mind the possible applications and advantages in the real world. With minor modifications and upgrades, we could use it in large social media platforms to bridge the gap between potential customers and the business owners.

1.4 Methodology

The proposed system does work in the following parts:

- Data preprocessing of the raw data
- Feature selection on the processed data
- Training and testing of models
- Advertisement prediction

(A) Data preprocessing

We did filtering data and defining which columns will be used for recommendation algorithms. In most of the cases, we have not used the rows which have any kind of reviews or purchases and no Ads content while in some cases, NaNs have been replaced with suitable values. The columns which are irrelevant have been dropped so that they do not affect predictions.

(B) Feature selection

In this system, we select the most relative feature by generating the score of each feature and selecting the most relevant one. We have columns like Rating, Reviews, Legitimate (True/False), didPurchase (True/False) which are very much effective in our Adanalysis. Since,We analyze user Comments along with the Ad content and make the ML model learning together of this to make it more efficient.

(i) Bag of Words

In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping

multiplicity. In the given dataset we considered the column “text” which contains the reviews given by a user for a product to achieve this concept. Using regular expression, we search a pattern and clean the data by removing the punctuation marks and special characters. By importing stop words from natural language toolkit, we later try to segregate the important and frequent used terms from the familiar words of English dictionary. We used Count Vectorizer to count the number of times a word occurs in a corpus. Hence, we display a list of words with their counts which are important and frequent in a corpus.

(ii) TD-IDF

TF-IDF stands for "Term Frequency, Inverse Document Frequency." It's a way to score the importance of words (or "terms") in a document based on how frequently they appear across multiple documents or corpus. If a word appears frequently in a document, it's important. Give the word a high score. But if a word appears in many documents, it's not a unique identifier. Give the word a low score. Therefore, common words like "the" and "for," which appear in many documents, will be scaled down. Words that appear frequently in a single document will be scaled up.

1.6 Organization of the Report

This section gives the overall picture of many chapters in this report.

Chapter 2 gives an overview of the project domain which describes the assumptions and the dependencies, user characteristics, functional requirements and constraints of the project.

Chapter 3 gives a detailed description of the software requirements and specification of ad analysis and classification using machine learning

Chapter 4 gives a detailed description about the high level design and the dataflow between the modules with the diagrams and detailed explanation.

Chapter 5 gives the detailed description of the system using a sequence chart.

Chapter 6 gives detailed description about implementation of ad analysis and classification using machine learning.

Chapter 7 gives the detailed description about Software testing.

Chapter 8 gives detailed description on the experimental results of ad analysis and classification using machine learning.

Chapter 9 gives clarity on the conclusion with this project and the scope for future enhancements.

1.7 Summary

This chapter deals with the introduction to the topic. The existing system for ad analyser is discussed. Research work, study material and other applications built on similar lines are discussed in this chapter. It also discuss in detail about the motivation, problem statement, objectives of the project.

CHAPTER-2

OVERVIEW OF ALGORITHMS USED

2.1 Introduction

The dataset which we are using is from the different ecommerce websites with 100k rows and 25 columns. We have an E- commerce dataset with details of Product like EAN number, Product Id, Name, Category, Date Added, Date Updated, Rating, Product URL, Product Manufacturer etc. Dataset also contains user information like User Id, Username, User Review for product, User Rating for product, User City, User Purchase and User Recommendation for product. User recommendation columns consist of True/False value to show if the user recommends a product or not. Rating consist of value between 1 to 5. The E- commerce dataset has approximately 72K rows and 25 columns after filtering and removing rows which are cannot be used by performing Exploratory Data Analysis and Data Cleaning.

In Algorithms used, we split dataset into two partitions: Test and train dataset by sampling in the ratios 20% and 80% respectively.

We aim to achieve the following for our system:

1. Suggest: Provide related items for the users from relevant and irrelevant collection of items.

2. Predict: Given a data of items purchased by Customers, we are trying to predict items for the user which can be useful for them based on user's past purchase history and location of the user.

3.Forecast: Demand forecasting can also be made of items according to the item sold in a country/continent

2.2 Python Programming Language

Python is a high-level functional language of programming used for general purpose computing. It was developed in 1991, and published. Python's Design philosophy is high readability of code. It provides constructs that allow easy interface programming for small as well as large projects.

Python refers to a device of adaptive sort, utilising automated memory management. Python has expanded its support since its introduction to various programming paradigms, such as object-oriented programming, imperative, procedural, and interactive, providing a broad and

extensive standard library.

2.3 Machine Learning Algorithms

Gaussian Naïve Bayes

Naive Bayes algorithm based on Bayes' theorem with the assumption of independence between every pair of features. This classifier work well in many real-world situations such as document classification and spam filtering. It requires a small amount of training data to estimate the necessary parameters and are extremely fast compared to more sophisticated methods. However, it is known to be a bad estimator.

2.4 Natural Language Processing (NLP):

Natural Language Processing, or NLP for short, is broadly defined as the automatic manipulation of natural language, like speech and text, by software. NLP is a collective term referring to automatic computational processing of human languages. This includes both algorithms that take human-produced text as input, and algorithms that produce natural looking text as outputs. We have analyzed the Review column of our dataset to understand customer reviews for each product. We have done basic pre-processing for the text such as removal of special characters (@,#), digits, punctuations, stopwords.

a) Bag of Words

In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity. In the given dataset we considered the column "text" which contains the reviews given by a user for a product to achieve this concept. Using regular expression, we search a pattern and clean the data by removing the punctuation marks and special characters. By importing stop words from natural language toolkit, we later try to segregate the important and frequent used terms from the familiar words of English dictionary. We used Count Vectorizer to count the number of times a word occurs in a corpus. Hence, we display a list of words with their counts which are important and frequent in a corpus.

b) TD-IDF

TF-IDF stands for "Term Frequency, Inverse Document Frequency." It's a way to score the importance of words (or "terms") in a document based on how frequently they appear across multiple documents or corpus. If a word appears frequently in a document, it's important. Give the word a high score. But if a word appears in many documents, it's not a unique identifier. Give the word a low score. Therefore, common words like "the" and "for," which appear in many documents, will be scaled down. Words that appear frequently in a single document will be scaled up.

c) Hashing

Hashing vectorizer applies a hashing function to term frequency counts in each document or corpus. Hash functions are an efficient way of mapping terms to features; it doesn't necessarily need to be applied only to term frequencies but that's how Hashing Vectorizer is employed here. Depending on the use case for the word vectors, it may be possible to reduce the length of the hash feature vector (and thus complexity) significantly with acceptable loss to accuracy/effectiveness (due to increased collision). If the hashing matrix is wider than the dictionary, it will mean that many of the column entries in the hashing matrix will be empty, and not just because a given document doesn't contain a specific term but because they're empty across the whole matrix. If it is not, it might send multiple terms to the same feature hash - this is the 'collision'.

2.5 Stacked Ensemble Model

Ensemble modeling is the process of running two or more related but different analytical models and then synthesizing the results into a single score or spread in order to improve the accuracy of predictive analytics and data mining applications. Stacking is a way of combining multiple models, that introduces the concept of a meta learner. Applying stacked models to real-world big data problems can produce greater prediction accuracy and robustness than do individual models. The model stacking approach is powerful and compelling enough to alter your initial data mining mindset from finding the single best model to finding a collection of good complementary models. Of course, this method does

involve additional cost both because you need to train a large number of models and because you need to use cross validation to avoid overfitting.

2.6 Summary

By building this Analysis system, we are able to find products which are similar and can be recommended to a set of users who have similar buying patterns through a set of ads generating based on the current analysis. After using various algorithms, we concluded that, Gaussian Naïve Bayes, NLP technique are most suitable for accurately predictions. NLP using Bag of Words is very much suitable for our dataset and for performing the required sentiment analysis serves as a way to let us know products demand and the kind of Ad based on them.

CHAPTER-3

SOFTWARE REQUIREMENTS AND SPECIFICATIONS OF AD ANALYSIS USING MACHINE LEARNING

CHAPTER 3

Software requirements and specification of ad analysis

Software requirement specification chapter explains the step by step development of the project. The functional prerequisites explain what the project must output and the non-functional prerequisites incorporate the limitations of the system.

3.1 Overall Description

In this section all the factors affecting the functioning of the system are described. The final developed model gives the relevant results.

3.1.1 Product Perspective

The Product should be easily maintainable and the product should be easy to use. The system consists of several modules which perform different tasks.

The final project should be easy to use and maintain. The result should be accurate which the purpose of the project is.

3.1.2 Product Functions

The main function of the project is to analyse and classify the advertisement. The product takes the inputs from the user. Once the input is taken the input is tested against the model which is already pre-processed and trained.

3.1.3 User Characteristics

This product analyses and classifies advertisements .This product has better accuracy than the existing models. Once the inputs are given then the analysis and classification is done. Since the final products aim is analysing and classifying this product can be used in different fields also.

3.1.4 Constraints and Dependencies

Ad analysis and classification using machine learning has few dependencies .The dependencies that the project has are:

- All the libraries should be updated
- The environment setting should be set properly
- The number of input variables is one of the main constraints in this system.

3.2 Specific Requirements

This part contains the prerequisites of the product which help the creators to create a framework. The type of ads and client qualities are not the major prerequisite .The major prerequisite would be the item should function regarding client accommodation. The last framework should consists all the required prerequisites.

3.2.1 Functional Requirements

The functional requirements of the system are:

- The database must be updated at each step.
- Data is reviewed and raw data should be sent for pre-processing.
- The pre-processed data after the feature selection is classified into two groups training and testing.

3.2.2 Performance Requirements

The proposed method takes inputs from the user and analysis and classifies the data, thereby providing high accuracy.

Performance requirements include the following:

- The Model should produce accurate results
- The expectation from the model is to analyse and classify the data
- The efficiency of the system should be more.

3.2.3 Supportability

The application is built using Jupyter notebook scripting tool which is based on python.

3.2.4 Software Requirements

The different software requirements of the application are as follows:

- Operating system: Windows 8 and above.
- Coding Language: Python 2.7 and above.
- Scripting tool: Jupyter notebook.
- Libraries: Pandas, scipy, numpy, matplotlib, etc.

3.2.5 Hardware Requirements

The following describe the hardware requirements for ideal running of the application.

- System: Pentium IV 2.4 GHz.
- Ram : 4 GB
- Any desktop / Laptop system with above configuration or higher level

3.2.6 Design Constraints

The main difficulties experienced are time taken to train the data and testing it. Another difficulty is to get accurate results in short time .To get more accurate data the information should be passed properly.

3.2.7 Interfaces

This section describes the interfaces. The two types of interfaces involved are User Interfaces and Software Interfaces.

3.2.7.1 User Interfaces of the system

The following are some of user interfaces options provided in the system:

- A form is presented to a new user to join.
- A login page to authenticate user credentials.
- An analysis page where the user can choose either the ECG analysis or the one based on his/her health data.
- A page with the result

3.2.7.2 Software Interfaces of the system

Ad analysis and classification is programmed using python. The analysis phase is done using python. The application is done on Jupyter notebook .

3.2.8 Non-Functional Requirements

The requirements that specify criteria that can be used to evaluate the operation of the system are:-

- Efficiency: System should perform all operations in short period of time.

- Availability: System must be up and running at all times and must accommodate a multitude of users.

- Reliability: Reliability based on this system defines the evaluation result of the system .The system should produce accurate results.

- Speed: The system must have a short latency period and must be responsive.

3.3 Summary

The requirements that must be remembered while building the application are defined in this section. These requirements are incorporated to get accurate results. All the requirements must be overseen while building and running the framework.

CHAPTER-4

HIGH LEVEL DESIGN OF AD ANALYSIS USING MACHINE LEARNING

CHAPTER 4

High Level Design of Ad Analysis using Machine Learning

Design is one of the most important steps in the development process. By having a suitable design model, we can have a model which an optimised and feasible solution to the problem. Using a high level design, we can be able to know the overview of the project, to able to break down the project into smaller sub parts or modules. In our project Ad Analysis using Machine Learning, we have number of modules designed which solves the problem in an as optimised way as possible.

Detailed design and with all the diagrams of all the modules have been described in this module.

4.1 Design Considerations

There are some few design considerations which should be followed while designing a project like the sequence which needs to be followed and how the interaction b/w the modules should be. So, there are a couple of design consideration which been considered in our project. And the given consideration has been described below.

4.1.1 General Constraints

There are a couple of general constraints which has been followed in our project. They are:-

- The data in which we are training the model should be accurate.
- Whenever we fetch the data through an API, the entire data in the .csv file should be fetched in one fetch operation.

- Whenever there is any missing value in the dataset, it should be replaced by any value.
- Whenever we find a row with irrelevant information, we discard the rows.
- The datasets considered in this project, should be large. In our project, we have a dataset which has 25 columns and 100,000 rows.

4.1.2 Development Methods

There are some development methods we have considered in our project are:-

- We have followed in sequential development in which an order was assigned to the modules, and development has been done in that sequence.
- We have implemented bottom-up approach in which from the bottom the individual modules we designed and development first and then integration and interaction b/w the modules were done to implement the system.

So, these are the development methods which were followed while implementing this project.

4.2 Architectural Strategies

There are some architectural strategies we have considered in our project:-

- The architectural strategies in this project, the architectural model will be given below. And the model is segregated into segments. And all the segments are been implemented in the framework. And a detailed architectural strategies is given below in the below sub chapters.

4.2.1 Programming Language

The programming language used in this project is python. As python is very commonly used for machine learning. And it is very easy to implement in projects especially in machine learning and data science domain. And it also supports a wide variety of libraries such as Pandas, numpy, scipy, matplotlib etc. which is required in this project. And the version of python used is 2.7 or higher so, it is easy to be compatible with other dependencies.

4.2.2 Data Storage Management

The data storage in this project is done in the excel sheet. It is done in the .csv format. The datasets is stores sequentially in the excel sheet format. With the help of API we are able to retrieve the data. Through one API call we should be able to retrieve all the data. So, this one of the objective while designing our project as it makes our project fast and increases the chances of accuracy and increases dataset coverage.

4.3 System Architecture

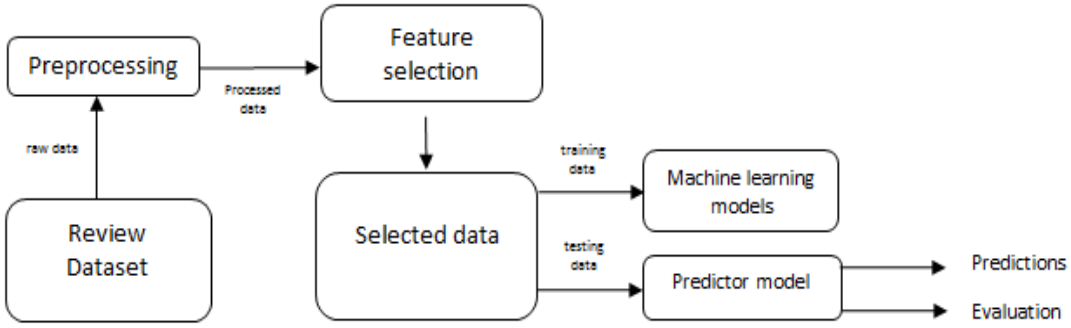


Fig 4.1 System Architecture

The above figure shows the system architecture which has the following modules and a detailed explanation of the fig. will be given below.

1. Review Datasets – This is the first module, in which the datasets are collected as many sources as possible, so that more number of computations can be done and we get more accurate results. In our project we have collected the data from amazon.com which has 100,000 rows and 25 columns.

2. Data Pre-processing – This is the second step in the process, where the data collected has to be cleaned. In this we are checking the any data in the rows is missing, if it is

missing we are filling data to that row and if any rows contain any irrelevant information, we discard/drop that row.

3. Feature Selection - This is the third step in the process where, from the datasets the feature is selected. In our project, we have ads as our datasets and we have tried to fetch the features based on the ads such as happy or sad etc. or any such feeling.

4. Machine Learning models – This is the third step in the process where the process where after selecting the feature. Machine learning models are incorporated like Gaussian and NLP algorithms are used to process and get the required result. The predicted output is obtained and evaluated.

So, this is the detailed design and description about the modules and the system architecture of the project.

4.1 Data flow diagrams

Data flow diagrams are the diagrams which provide a data flow b/w the entities. So that user or the client will be able to vision the full working of the system. There are three levels in the DFD design DFD 0, DFD 1 and DFD 2. The working explanation with examples will be given below.

4.1.1 Data flow diagrams – Level 0

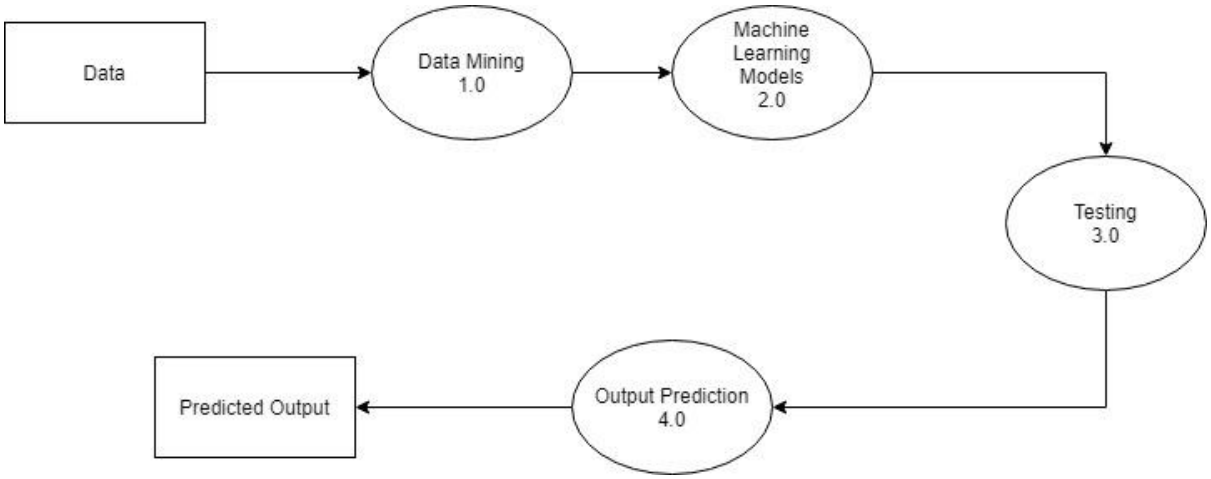


Fig 4.2 DFD Level 0 of Ad Analysis Using machine learning

DFD Level 0 is the diagram which represents overview data flow to the project. It is represents in a simple diagram how the data flow happens in a module. So, this is the DFD level 0 where the data flow takes place in the following stages before predicting the right output. So, the in detailed data flow will be explained in DFD Level 0.

4.4.2 Data flow diagram – Level 1

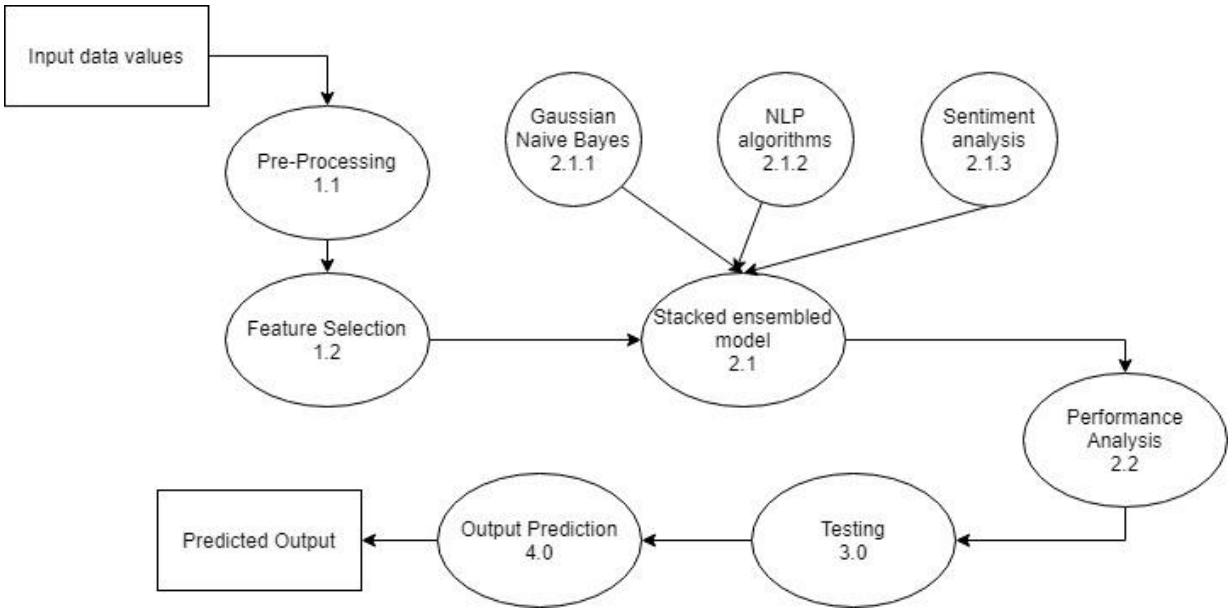


Fig 4.3 DFD Level 1 of ad analysis using machine learning

DFD Level 1 shows how the modules are divided into sub modules and how the data flow is given in more detailed way. It represents the interaction with the various sub modules and the data flow b/w the sub modules from one end to another end. So, in the figure 4.3 the data flow is shown in detail and shows the full working and data flow in the system. So, this is about DFD Level 1.Module 3.0 will be explained in detail in DFD Level 2.

4.4.3 Data flow diagram – Level 2

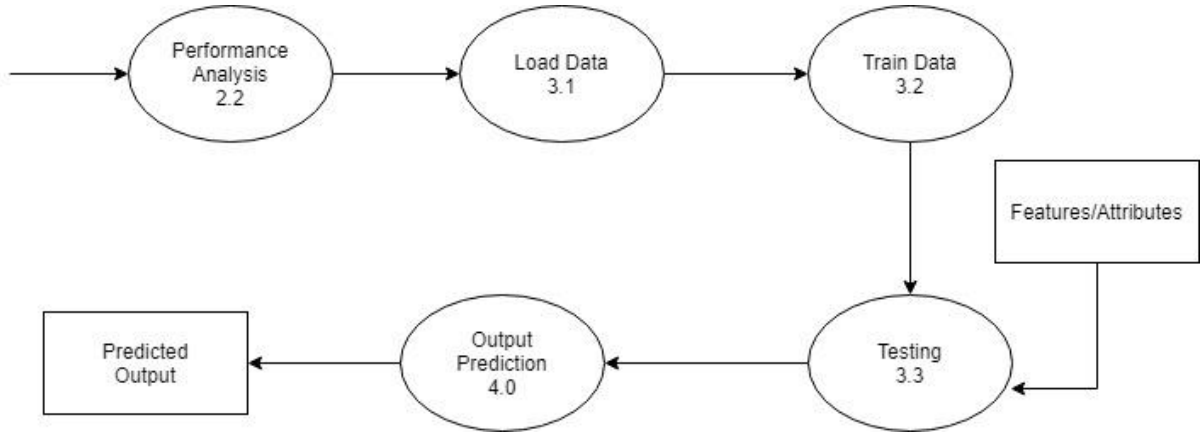


Fig 4.4 DFD Level 2 of Ad Analysis using machine learning

DFD Level 2 is used to represent the data flow in detail in each module and sub module. It represents the total working of the module 3.0(Testing). In Fig 4.4 it shows the working and data flow of a module testing where after the system is trained, computation is provided on the test to get the predicted output. So, the DFD Level 2 explains the following module.

4.5 Summary

In this chapter, the information given in this chapter, it is represent the high level design and the data flows b/w the modules and the interaction b/w each. In this chapter, we also learnt about the architecture and how the system is designed and interaction with each other. All the above sections are outline of these chapters.

.

CHAPTER-5

DETAILED DESIGN OF AD ANALYSER SYSTEM

DETAILED DESIGN OF AD ANALYSER SYSTEM

In the detailed design phase, the internal logic of every module specified in the high-level design is determined. Specifically, in this phase the design of each module, the low-level components and subcomponents are described. Each module input and output type, along with the possible data structures and algorithms used are documented during the detailed design phase. The following sections provide such information of the modules.

5.1 Sequence Chart

The sequence chart shows the control flow among the modules in the system. It explains all the identified modules and the interaction between the modules. It also explains the identified sub-modules. The sequence chart shown in fig 5.1 explains the input for each module and output generated by each module.

• Input Module: The input module contains the processed data from the dataset which is cleaned and extracted.

• Feature extraction module: We select the most relative feature by generating the score of each feature and selecting the most relevant.

• Output module: Keeping the independent and dependent variables same across various algorithms, we try and find acceptable results of various algorithms used for our E-commerce dataset

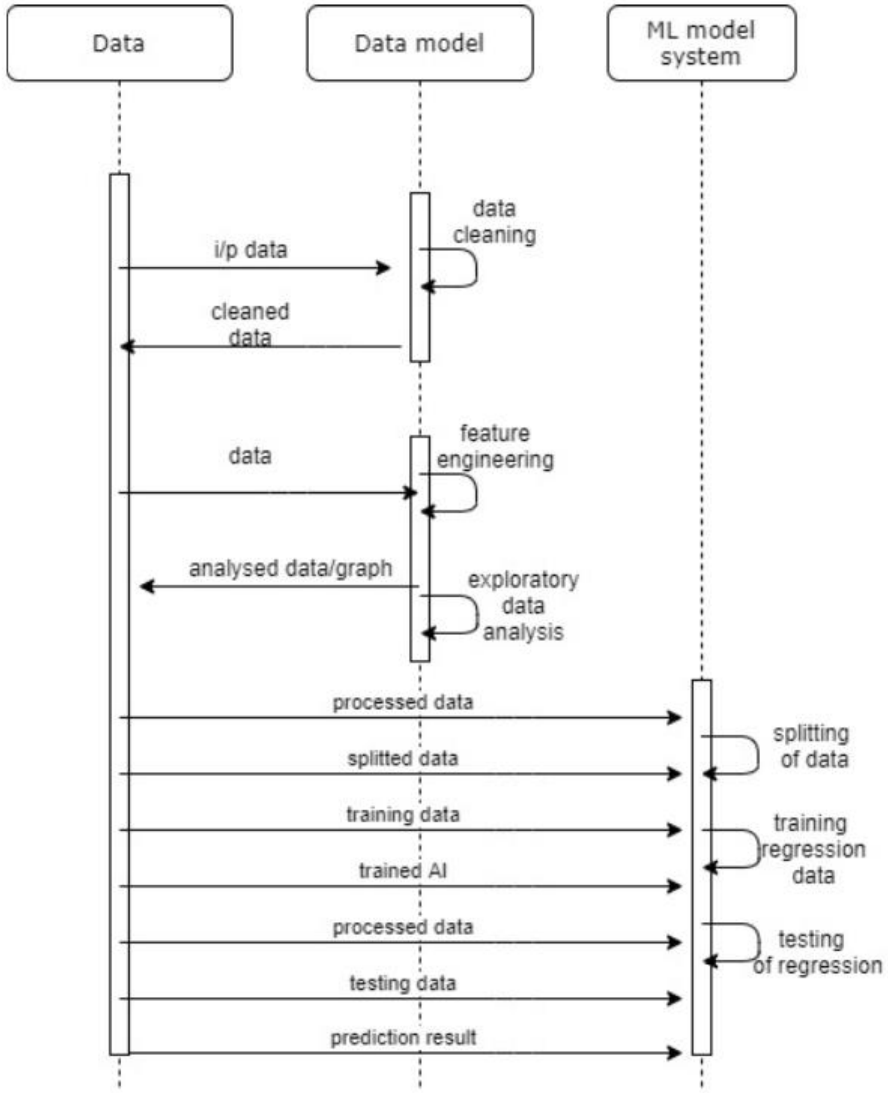


Fig 5.1 Sequence chart of ad analyser

5.2 Summary

The internal working of the applications' two modules with the necessary data flow through each of them has been described in this chapter. A clear view on control flow within system was conveyed by the structure chart with the functionality of its modules being explained.

The flow charts explain the working of each module with flow of control in the module specified which gives a complete understanding of the functioning.

CHAPTER-6

IMPLEMENTATION OF AD ANALYSIS USING MACHINE LEARNING

CHAPTER 6

Implementation of ad analysis using machine learning

The implementation stage is an important phase in the project as it contains the major part of the project that help in achieving the goal. In this stage the low level designs are changed into programs which help us achieve the goal. This stage involves genuine implementation of thoughts that were depicted in the analysis and design stage. The program should be strategized in way so that it can help in achieving reusability.

6.1 Programming Language Selection

The programming language selected to implement the project is python.

Python is a very easy to understand, simple and expressive language. It is also highly interpreted, simple, easy and highly readable program syntax and user friendly. Python has extensive support for web applications. Python has huge number of packages and libraries that can be used for different purposes.

6.2 Platform Selection

Testing was done on a test node to ensure no incompatibility issues among systems. The major part of the coding in done in Jupyter notebook which supports python.

6.3 Code Conventions

This part explains the coding standards followed in this project. Legitimate coding standards ought to be followed in light of the fact that huge undertakings ought to be coded in a steady style. The code standards are followed as they are crucial as they help in debugging and also allows the reusability of code.

6.3.1 Naming Conventions

The use of naming convention helps in code readability which helps in debugging. Naming conventions should be followed while naming packages, script, variables, methods such that they are logical and relatable. The project uses Python and the naming convention followed is described below:

The conventions followed in Python for this project are as follows:

- **Classes**: The name of the class should be a noun with the first letter of the name being in capital. Nonetheless, all the built in classes of python are lowercase.

  Example: . stopwords. word()

- **Methods**: The name of methods should be a verb describing the function of method and it is all lowercase. Words in a method name should be separated by an underscore. The method name is a description of its function.

  Example: removalofwords( ).

- **Variables**: Variable names must be short, meaningful and if multiple words are present, they are connected by underscore. Instance variable names should be all lowercase.

  Example: train, ad.

6.3.2 File Organization

In the backend Python services side, the files are arranged as follows:

- The trained model is present in the src folder.
- The Python codes are stored in the source folder of the application. In the src folder, there are multiple sub directories, including mitdb, templates,static and uploads.

6.3.3 Declarations

There are some standards to be followed while coding in python language. Logical names are given to the variables and methods used which make it easy to understand the role of each entity declared. All the variables are declared in different lines.

6.3.4 Comments

The comments in python are a good feature, it is not compulsory to include comments, these help in understanding the code. Some other person can understand the code well if comments are used. This also helps in debugging the code. In Python comments start with a '#'.Multiline comments are not allowed in python.

6.5 Summary

This chapter contains the detailed explanation of programming language, its coding standards. It also explains about the naming convection used during the development.

CHAPTER-7

SOFTWARE TESTING OF AD ANALYSIS USING MACHINE LEARNING

CHAPTER 7

Software Testing of Ad Analysis using Machine Learning

Software Testing is to recognize bugs or errors by performing tests on each module of the projects independently. All the modules are joined to form a frame work, testing is done on this framework. Before testing, all the functional requirements should be satisfied. The test cases are picked so that all the framework functionalities are tested. The test cases are chosen from the inputs which have some output, some test cases show a message. In the testing phase of the project unit testing of all the modules are performed first and then some modules are integrated and integration testing is done, finally full system testing is done to ensure proper functioning.

In this section, distinct test cases are intended for testing the functionality of all modules. Test cases are executed under similar conditions as the model. It checks for the functionality of all modules. Once the testing is done the model will be ready to be used in real-time.

7.1 Test Environment

The application analyses and classifies the advertisements. The output is viewed through the Jupyter notebook. Integration and system testing were done on a test node which sets up a simulated product environment.

Test Environment:

- OS: Windows/MAC
- Tool: Jupyter Notebook
- Dataset: Data from E-commerce websites

7.2 Unit Testing

Unit testing is done to check whether the function is returning the expected output. Every module is tested separately to ensure the functionality .This helps in locating the errors accurately. This section describes the unit tests run with test case details and explanations.

The Table7.1 below represents the test case to identify the number of characters/stop words/ uppercase/words/special characters on which this testing is performed.

Table 7.1 Identification of characters/stop words/words

Sl .No	1
Test Case	Identify the number of characters/stop words/ uppercase/words/special characters
Feature	Identification of characters
Description	Test to identify he number of words/characters
Sample Input	Data from E-commerce site
Expected Output	Identify necessary elements
Actual Output	Same as expected
Remark	Successful

Table 7.1 represents the test case details for testing identification of number of character/words module. Test was conducted successfully. The input given was the data from a e-commerce website which was successfully processed by the sub module.

Data processing is tested to ensure smooth running in all conditions. This is described in Table 7.2.

Table 7.2 Data Processing test

Sl .No.	2
Test Case	Data Processing
Feature	Removal of punctuation/ stop words/rare words / spelling correction
Description	Test to remove the punctuation, stop words, common words, spelling correction.
Sample Input	Data after the identification of number of words/characters
Expected Output	Data after the removal of punctuation ,stop words and common words
Actual Output	Same as expected
Remark	Successful removal

Table 7.2 shows the test case details of removal of stop words/common words/punctuation. The input is the data after the identification of common words/stop words/characters. This test was successful.

Table 7.3 Stacked ensemble algorithm

Sl No.	3
Test Case	Stacked ensemble algorithm
Feature	Working of algorithms
Description	Logistic regression, Random forest tree, K -nearest neighbours.
Sample Input	Pre-processed data
Expected Output	Algorithm training and testing of data
Actual Output	Same as expected
Remark	Successful

Table 7.3 represents the test case for stacked ensemble algorithm. The input to this module is pre-processed data. This test was also successful.

7.3.2 Integration Testing

The integration testing includes the testing of all the modules together. This is shown in Table 7.4.

Table 7.4 Feature Engineering

Sl No.	4
Test Case	Feature Engineering
Feature	Tokenization , Stemming , N-grams , Term frequency
Description	Appropriate working of all the integrated modules
Sample Input	Pre-processed data
Expected Output	The pipeline of algorithms executed on the processed data
Actual Output	Same as expected
Remark	Successful

Table 7.4 represents the testing of feature engineering. The tests performed were accurate and satisfy the requirements of the function.

7.4 System Testing

System testing validates the functionality of the entire application. System testing should be done so that all the unit modules are associated properly. This is used to remove any errors from the system and to improve the quality of project.

System testing is represented in Table 7.6.

Table 7.6 System testing

Sl No.	6
Test Case	System testing
Feature	Full system testing of ad analysis and classification using machine learning
Description	Checks complete working of the full system
Sample Input	Data from the e-commerce website
Expected Output	Analysis and Classification of Advertisements
Actual Output	Same as expected
Remark	Successful

Table 7.6 represents the system testing. All the modules are combined and tested. The system should analyse and classify the advertisements. All the functions are tested positively.

7.5 Summary

This section consists of the testing methods from the individual unit testing of all the modules to combining the modules and performing integration testing. System testing ensures that all the modules are working as per their functionality and the system is error free.

CHAPTER-8

EXPERIEMENTAL RESULT AND ANALYSIS OF AD ANALYSIS USING MACHINE LEARNING

CHAPTER 8

Experimental Result of Ad Analysis and Classification using Machine Learning

In the analysis phase of project, experiments are performed to check the input to the process which frequently determines the output of the process. It helps in determining the better input set. Different metrics are deployed for contrasting the outcome obtained. This section tests the performance of the application.

8.1 Evaluation Metrics

Evaluation metrics provide the benchmark of the algorithm used in prediction. The efficiency of these algorithms can be decided by applying their metrics as a basis for comparison. In this project, the outcome achieved by providing contrasting inputs is compared to the optimal outcome as per functionality to analyse the satisfaction of metrics. The project was evaluated based on the accuracy of the system outputs.

In this project most relative feature is selected by generating score for each feature. There are Columns like Rating, Reviews, Legitimate (True/False), didPurchase (True/False) which are very much effective in our Ad analysis. Since both user comments are analyzed along with the Ad content this makes the ML model learning more efficient.

8.2 Experimental Dataset

The dataset which we are using is from the different ecommerce websites with 100k rows and 25 columns. An E- commerce dataset with details of Product like EAN number, Product Id, Name, Category, Date Added, Date Updated, Rating, Product URL, Product Manufacturer etc. is used. Dataset also contains user information like User Id, Username,

User Review for product, User, Rating for product, User City, User Purchase and User Recommendation for product. User recommendation columns consist of True/False value to show if the user recommends a product or not. Rating consists of values between 1 to 5. The E- commerce dataset has approximately 72K rows and 25 columns after filtering and removing rows which are cannot be used by performing Exploratory Data Analysis and Data Cleaning.

In Algorithms used, we split dataset into two partitions:

Test and train dataset by sampling in the ratios 20% and 80% respectively.

We aim to achieve the following for our system:

3. **Suggest:** Provide related items for the users from relevant and irrelevant collection of items.

4. **Predict:** Given a data of items purchased by Customers, we are trying to predict items for the user which can be useful for them based on user's past purchase history and location of the user.

8.3 Performance Analysis

Gaussian Naïve Bayes	94.69%
Natural Language Processing for Sentiment Analysis	Bag of Words TF-IDF Hashing, and sentiment analysis
Stacked Ensemble model	Voting Classifier ~94%

Fig 8.3 Performance measured

This is the performance measured was approximately ~94% as the final measures. There are two algorithms which were followed to implement this application. First one, was the Gaussian naïve Bayes to dependence b/w the data pair, and the accuracy was found to be

94.69%. And then NLP algorithms were incorporated and 3 algorithms were used Bag of Words, TF-IDF and Hashing and sentiment analysis was done, where the accuracy was ~94%. And Stacked Ensemble model was used to combine both of the models and accuracy was ~94%.So, this is the performance measure which was obtained from the project.

8.4 Summary

The Ad analysis and classification works efficiently and After using various algorithms, we concluded that, Gaussian Naïve Bayes, NLP technique are most suitable for accurately predictions. NLP using Bag of Words is very much suitable for our dataset and for performing the required sentiment analysis serves as a way to let us know products demand and the kind of Ad based on them.

CHAPTER-9

CONCLUSION AND FUTURE ENHANCEMENT

CHAPTER 9

Conclusion and Future Enhancement

With the increasing number of ads, they have become one of the ways an organisation can make money. So, with this organisations find it difficult to show the ads to its customers which interest them. The aim of our project was to provide an optimal solution which will an organisation or any customer for recognising the ads and display it to the customer's according to their interest. So, in this project we have used machine learning algorithms to build this application. We have used naïve Bayes Gaussian algorithm to find the independence b/w each pairs of datasets. And we have used natural language processing algorithms to find the information about the ads and how is it interesting the user. After this step, sentiment analysis was done and finally stacked ensemble model was incorporated in our project, where we combine two or model to get better result. So, after using this model we get accuracy up to 94%.

So, to conclude from this project, we can have an application which provides optimal solution of recommending ads with less data as possible. And the application which is built is scalable to more number and more number of customers. So, that it is a win-win situation for both customers and the organisation.

9.1 Limitations of the project

Although, the project which is developed has successfully passed all the criteria and standards which were set but there are a few the improvements can be made and as of now they are limitations. They are:-

- The data resources in this domain are very less. There are not many resources available which provides information about this domain.
- Even though, an optimal model has been built this provides a solution to very less data as possible. There is a threshold of the amount of data below which the model becomes unsuccessful.

- GUI was not developed during this project.

So, these are the limitations of this project and further can be worked upon in increasing the efficiency and accuracy of the project.

9.2 Future Enhancements

There are some of the limitations mentioned above which can be taken as future works. And there is a high chance of possibility for future enhancements which can be done in this project. And some of them are mentioned below:-

- A GUI can be developed in this project.
- This can be integrated with software's such as Facebook, Instagram or any website and increase the sales of a particular product.
- There is a chance of improving the model in such a way that with the help of less datasets we can be able to improve the accuracy.

So, these are the future enhancements which can be worked upon in the future and making it a better product.

9.3 Summary

This section provides a detailed information and a clear understanding about the conclusion of the project on what was the goal and we tried to achieve and the limitations of the project and future enhancements which can be worked on and increase the efficiency and accuracy and provide a better product.

References

[1] Huitao Luo, "Algorithms for Video Object Detection and Segmentation with Application to Content-Based Multimedia Systems" , Columbia University, 2000.

[2] Vladimir Pavlov, Vladimir Khryashchev, Evgeny Pavlov, Lev Shmaglit " Application for Video Analysis Based on Machine Learning and Computer Vision Algorithms ", Yaroslavl State University Yaroslavl, Russia, .

[3] Francesco Camastra, Alessandro Vinciarelli ," Machine Learning for Audio, Image and Video Analysis".

[4] Zhong Guo,"Object Detection and Tracking in Video",Department of Computer Science, Kent State University, 2001.

[5] Nevenka Dimitrova, Hong-Jiang Zhang, Behzad Shahraray, Ibrahim Sezan, Thomas Huang, Avideh Zakhor, " Applications of Video-Content Analysis and Retrieval", IEEE 2000.

[6] R. Castagno, T. Ebrahimi, and M. Kunt. Video segmentation based on multiple features for interactive multimedia applications. IEEE Transactions onCircuits and Systems for Video Technology, 8(5):562{571, 1998.

[7] E. Chalom. Image segmentation using multi-dimensional attributes. Thesis Proposal, MIT Media Lab, 1996.

[8] E. Chalom and V.M. Bove Jr. Segmentation of an image sequence using multidimensional image attributes. In IEEE International Conference on ImageProcessing, Lausanne, Sept. 1996.

[9] S.-F. Chang, W. Chen, H.J. Meng, H. Sundaram, and D. Zhong. A fully automated content-based video search engine supporing spatiotemoral queries. IEEE Transactions on Circuits and Systems for Video Technology, 8(5):602{615, 1998.

[10] R. Chellappa, C.L. Wilson, and S. Sirohey. Human and machine recognitionof faces: A survey. Proceedings of IEEE, 83(5), 1995.L. Chiariglione. Statement on MPEG receiving the Emmy Award. MPEG 96, Oct. 2, 1996.

[11] T.C Chiueh, T. Mitra, and C.K. Yang. Zodiac: a history-based interactivevideo authoring system. In ACM International Multimedia Conference, Bristol, England, Sept. 1998.J.G. Choi, S.W. Lee, and S.D. Kim. Spatio-temporal video segmentation usinga joint similarity measure. IEEE Transactions on Circuits and Systems forVideo Technology, 7:279{285, 1997

[12] R. Ciapini, L. Blanc-Feraud, M. Barlaud, and E. Salerno. Motion-based segmentation by means of active contours. In IEEE International Conference onImage Processing, Santa Barbara, CA, 1997.

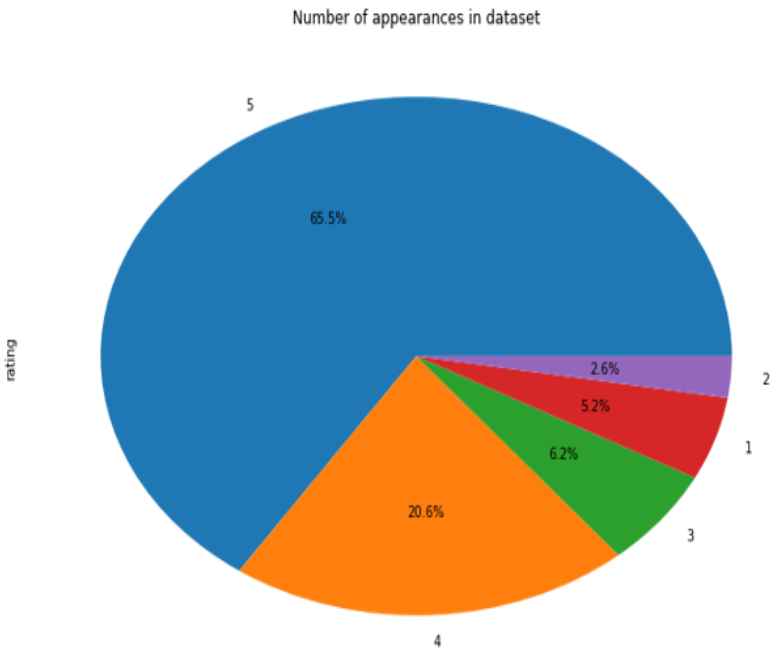
[13] S. Colonnese, A. Neri, G. Russo, and P. Talone. Moving objects versus still background classication: a spatial temporal segmentation tool for MPEG-4.ISO/IEC JTC1/SC29/WG11, MPEG 96/571, 1996.

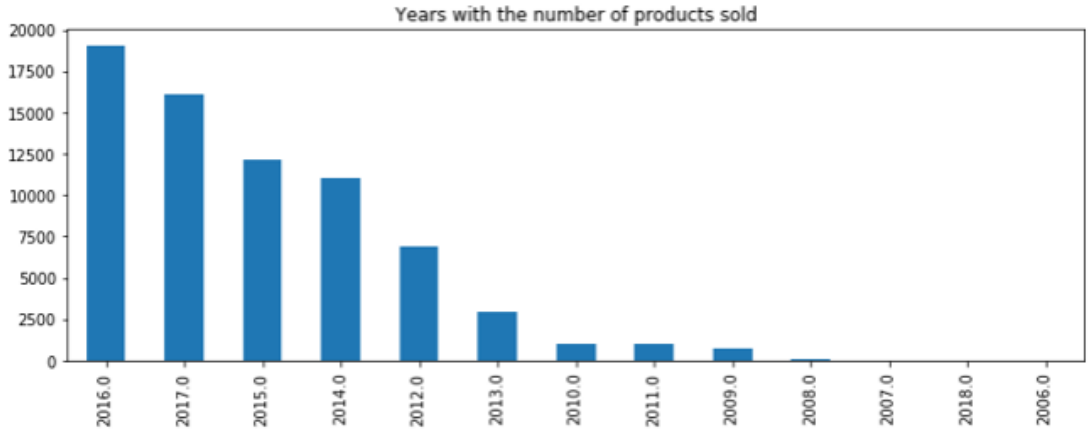
[14] T.F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active shape models,their training and applications. Computer Vision and Image Understanding, 61:38{59, Jan. 1995.

[15] T.H. Cormen, C.E. Leiserson, and R.L. Rivest. Introduction to Algorithms, chapter 25.2. MIT Press, 1990.

Appendix -1

(Screenshots)




Years with the number of products sold
20000
17500
15000
12500
10000
7500
5000
2500
0
2016.0
2017.0
2015.0
2014.0
2012.0
2013.0
2010.0
2011.0
2009.0
2008.0
2007.0
2018.0
2006.0

Basic Pre-processing

Lower case

Convert all the words in the reviews to lowercase for processing

In [109]:
```
train['text'] = train['text'].apply(lambda x: " ".join(x.lower() for x in x.split()))
train['text'].head()
```

Out[109]:
```
0    i love this album. it's very good. more to the...
1    good flavor. this review was collected as part...
2                                          good flavor.
3    i read through the reviews on here before look...
4    my husband bought this gel for us. the gel cau...
Name: text, dtype: object
```

Removing Punctuation

Remove any punctuations present in the reviews for processing

In [111]:
```
train['text'] = train['text'].str.replace('[^\w\s]','')
train['text'].head()
```

Out[111]:
```
0    i love this album its very good more to the hi...
1    good flavor this review was collected as part ...
2                                          good flavor
3    i read through the reviews on here before look...
4    my husband bought this gel for us the gel caus...
Name: text, dtype: object
```

Removal of Stop Words

Stop words are words which are filtered out before or after processing of natural language data (text). We remove some of the most common words—including lexical words, such as "want" in order to improve performance.

In [113]:
```
from nltk.corpus import stopwords
stop = stopwords.words('english')
train['text'] = train['text'].apply(lambda x: " ".join(x for x in x.split() if x not in stop))
train['text'].head()
```

Out[113]:
```
0    love album good hip hop side current pop sound...
1          good flavor review collected part promotion
2                                          good flavor
3    read reviews looking buying one couples lubric...
4    husband bought gel us gel caused irritation fe...
Name: text, dtype: object
```

Common word removal

We also remove commonly occurring words from our text data.

In [115]:
```
freq = pd.Series(' '.join(train['text']).split()).value_counts()[:10]
freq
```

Out[115]:
```
great        20936
product      20247
movie        19729
review       18906
part         18665
promotion    17733
collected    17725
```

N-grams

N-grams are the combination of multiple words used together. Ngrams with N=1 are called unigrams. Similarly, bigrams (N=2), trigrams (N=3) and so on can also be used.

In [135]:
```
TextBlob(train['text'][0]).ngrams(2)
```

Out[135]:
```
[WordList(['album', 'hip']),
 WordList(['hip', 'hop']),
 WordList(['hop', 'side']),
 WordList(['side', 'current']),
 WordList(['current', 'pop']),
 WordList(['pop', 'sound']),
 WordList(['sound', 'hype']),
 WordList(['hype', 'listen']),
 WordList(['listen', 'everyday']),
 WordList(['everyday', 'gym']),
 WordList(['gym', 'give']),
 WordList(['give', '5star']),
 WordList(['5star', 'rating']),
 WordList(['rating', 'way']),
 WordList(['way', 'metaphor']),
 WordList(['metaphor', 'crazy'])]
```

Term frequency

Term frequency is simply the ratio of the count of a word present in a sentence, to the length of the sentence.

Therefore, we can generalize term frequency as:

TF = (Number of times term T appears in the particular row) / (number of terms in that row)

In [137]:
```
tf1 = (train['text'][1:2]).apply(lambda x: pd.value_counts(x.split(" "))).sum(axis = 0).reset_index()
tf1.columns = ['words','tf']
tf1
```

Out[137]:

	words	tf
0	flavor	1

Inverse Document Frequency

The intuition behind inverse document frequency (IDF) is that a word is not of much use to us if it's appearing in all the documents.

Therefore, the IDF of each word is the log of the ratio of the total number of rows to the number of rows in which that word is present.

IDF = log(N/n), where, N is the total number of rows and n is the number of rows in which the word was present.

In [139]:
```
for i,word in enumerate(tf1['words']):
  tf1.loc[i, 'idf'] = np.log(train.shape[0]/(len(train[train['text'].str.contains(word)])))

tf1
```

Out[139]:

	words	tf	idf
0	flavor	1	5.141692

The more the value of IDF, the more unique is the word.

Voting Classifier Ensemble

In [4]:
```
X=df[['didPurchase','rating']]
y=df['doRecommend']

clf1 = LogisticRegression(random_state=1)
clf2 = RandomForestClassifier(random_state=1)
clf3 = GaussianNB()
clf4 = KNeighborsClassifier(n_neighbors=10)
eclf = VotingClassifier(estimators=[('lr', clf1), ('rf', clf2), ('gnb', clf3),('knn',clf4)], voting='hard')
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=None)
```

In [5]:
```
eclf.fit(X_train, y_train)
```

Out[5]:
```
VotingClassifier(estimators=[('lr',
                              LogisticRegression(C=1.0, class_weight=None,
                                                 dual=False, fit_intercept=True,
                                                 intercept_scaling=1,
                                                 l1_ratio=None, max_iter=100,
                                                 multi_class='auto',
                                                 n_jobs=None, penalty='l2',
                                                 random_state=1, solver='lbfgs',
                                                 tol=0.0001, verbose=0,
                                                 warm_start=False)),
                             ('rf',
                              RandomForestClassifier(bootstrap=True,
                                                     ccp_alpha=0.0,
                                                     class_weight=None,
                                                     crite...
                                                     n_estimators=100,
                                                     n_jobs=None,
                                                     oob_score=False,
                                                     random_state=1, verbose=0,
                                                     warm_start=False)),
                             ('gnb',
                              GaussianNB(priors=None, var_smoothing=1e-09)),
                             ('knn',
                              KNeighborsClassifier(algorithm='auto',
                                                   leaf_size=30,
                                                   metric='minkowski',
                                                   metric_params=None,
                                                   n_jobs=None, n_neighbors=10,
                                                   p=2, weights='uniform'))],
                 flatten_transform=True, n_jobs=None, voting='hard',
                 weights=None)
```

Staked Ensemble

In [8]:
```
training, valid, ytraining, yvalid = train_test_split(X_train, y_train, test_size=0.3, random_state=0)
```

In [9]:
```
clf2.fit(training,ytraining)
```

Out[9]:
```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=None, oob_score=False, random_state=1, verbose=0,
                       warm_start=False)
```

In [10]:
```
clf4.fit(training,ytraining)
```

Out[10]:
```
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=10, p=2,
                     weights='uniform')
```

In [11]:
```
preds1=clf2.predict(valid)
preds2=clf4.predict(valid)
```

In [12]:
```
test_preds1=clf2.predict(X_test)
test_preds2=clf4.predict(X_test)
```

In [13]:
```
stacked_predictions=np.column_stack((preds1,preds2))
stacked_test_predictions=np.column_stack((test_preds1,test_preds2))
```

In [14]:
```
meta_model=LinearRegression()
meta_model.fit(stacked_predictions,yvalid)
```

Out[14]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [15]:
```
final_predictions=meta_model.predict(stacked_test_predictions)
```

In [ ]:
```

```

In [16]:
```
count=[];
y_list=y_test.tolist()
for i in range(len(y_list)):
    if (y_list[i]==np.round(final_predictions[i])):
        count.append("Correct")
    else:
        count.append("Incorrect")
```

In [17]:
```
import seaborn as sns
sns.countplot(x=count)
```

Out[17]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x19e00aff1d0>
```

In [18]:
```
accuracy_score(y_test,np.round(final_predictions))
```

Out[18]:
```
0.9561522933195624
```

Appendix -2

Ad Analysis using Machine Learning

Yashas C N[1], Kishan V[2], Pranay Reddy Juturu[3] and Prof. Pavithra H[4]

[1,2,3]*Department of Computer Science and Engineering, R.V College of Engineering, Bengaluru, Karnataka , India*
[4]*Professor, Department of Computer Science and Engineering, R.V College of Engineering, Bengaluru, Karnataka, India*

Abstract - *In this paper we understand the concept of recommendation system and how beneficial it is to understand and analyze the advertisements as per user. As the audience keep evolving and there are different generation of users who are interested in different type of content. So, it is really important to recommend the right type of ad's in this evolving system so that the situation can be a win-win situation for the user and for the ad agencies as well. By using right type of recommendation system, this provides a solution of dynamically providing the information to the user and helps in promoting the product to the user and increasing their sales and business. In this paper, we'll discuss and analyze the type of approach used in solving this problem.*

1. INTRODUCTION

Machine Learning is a field of computer science which uses statistical models to classify or predict the data .We are separating the data as test data and train data, where we use the train data to train the model and test data to test and see how that works and check the result meets the expectations and to improve the accuracy of the model.

In this paper, we have briefly explained a model which could recommend ads to the user in the most optimal way and recommends the content in the social media according to the user's interest and has a higher percentage of accuracy.

According to the recent study it is shown that, an average millennial spends around 8 hours a day on social media such (Facebook, Instagram etc..) and in YouTube around 4 billion videos are being watched every day. So social media, is a great source of obtaining the data. So, there are techniques incorporated by large corporations such as Facebook, Google who recommendation system to display the content according to the user's interest. But also, there are some websites by using the right type of recommendation system will be able to provide the right type of content to the right users.

2. RELATED WORK

There are lot of related work with respect to ad analysis using machine learning. One of the most used work is by video and sound analysis. Where the sound and video is segregated into frames and hashing is used so that there are no duplicates and naïve Bayes and sentiment analyzer is used for further recommendation. This method is not optimal as this method introduces to a lot of noise and accuracy will be low. The approach used in this journal is feasible as the accuracy will be high and solves the above problem.

3. ANALYSIS METHODOLOGY

Fig 1.1

In the fig 1.1 the architecture of the proposed model has been explained. The following steps have been explained to achieve this process. They are: -

3.1 Data Visualization

In this method, the data is obtained from different e-commerce website. The data 100k rows and 25 columns and has different fields. The dataset which is chosen 80% test data and 20% train data. The data visualization is shown in the Fig. 1.2 and Fig 1.3.

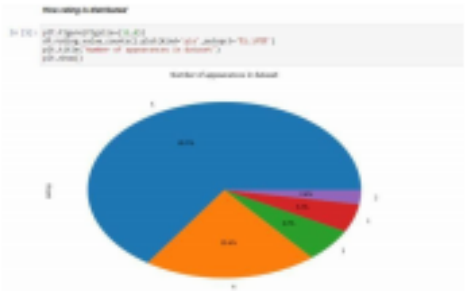


Fig 1.2. Distribution of Rating

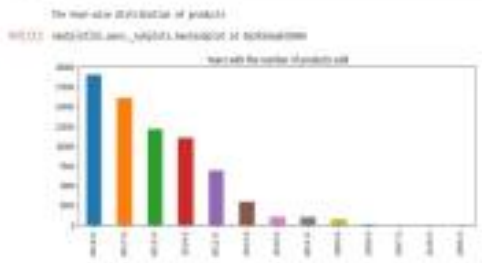


Fig 1.3. Year wise distribution of the product.

3.2 Data Preprocessing

Data pre-processing is the next step in this process as the data is obtained. There will be columns which have any kind of rows of content if there are rows with any type of non-relevant information, that particular row is dropped. And if there are NaN values, relevant information will be in those columns. The code snippet for this process is given below: -

Fig 1.4 Dropping the values which are not relevant

Fig 1.5 Fill Nan with suitable values.

3.3 Feature Selection

This is the next step in the process where various machine learning algorithms are incorporated to obtain the desired result: -

3.31 Gaussian Naive Bayes

Naive Bayes algorithm is based on Bayes' theorem with the assumption of independence with every pair of features. This requires extremely small amount of data to estimate the parameters but is fast compared to the more methods. However, it is a bad classifier but is used in real world application such as spam filtering and document classification.

Fig 1.6 Accuracy using Gaussian Naive Bayes

3.32 Natural language Processing (NLP)

Natural Language processing is a term which is used for automatic computation processing of human languages. This includes both algorithms that take human-produces text as input, and algorithms that produce natural looking text as input.

We have analyzed the review column for our dataset to understand customer review for each product. We have done preprocessing for removal special characters, digits, punctuations etc..

Fig 1.7 Removal punctuation and stop words.

This step (natural language processing) is achieved by 3 algorithms: -

A. Bag of Words.

B. TF-IDF.(Term Frequency, Inverse Document Frequency)

C. Hashing

The code snippet for each of these algorithm is given below: -

Fig 1.8 Bag of Words using CountVectorizer

Fig 1.9 TF-IDF using TfidfVectorizer

Fig 1.10 Hashing using HashingVectorizer

3.33 Sentiment Analysis

Sentiment is like a combination of tone of voice, word choice, and writing style all rolled into one. Natural language with labels about positivity or negativity, we can develop agents that can learn to understand any sentiment. The code snippet for sentiment analysis is given below.

Fig 1.11 Sentiment Analysis for the text

3.34 Stack Ensemble Model

Ensemble modeling is the process of running two or more related but different analytical models and then synthesizing the results into a single score or spread in order to improve the accuracy of predictive analysis and data mining applications. Stacking is a way of combining multiple models, that introduces the concept of a meta learner. Applying stacked models to real-world big data problems can produce greater prediction accuracy and robustness than do individual models.

Fig 1.12 Accuracy using Stacked Ensemble for Voting Classifier.

4. Results

Keeping the independent and dependent variables same across various algorithms, we found acceptable results of various algorithms used for our E-commerce dataset.

Dependent variable: doRecommend Independent variable: ProductId, didPurchase, Username, Rating.

Since we take into consideration various independent variables and find the correlation between them, our aim to predict the likelihood for the Advertisement being legitimate to the user and percentage if correctness.

Gaussian Naïve Bayes	94.69%
Natural Language Processing for Sentiment Analysis	Bag of Words TF-IDF Hashing, and sentiment analysis
Stacked Ensemble model	Voting Classifier –94%

5. Conclusion

By building this analysis system ,we are able to find products which are able to find the products which are similar and can be recommended to a set of users who have similar buying pattern through a set of ads generating based on the current analysis, After using various algorithms, we concluded that, Gaussian Naïve Bayes, NLP technique are most suitable for our dataset and for performing the required sentiment analysis serves as a way to let us know products demand and the kind of Ad based on them.

REFERENCES

[1] R Vinit Kaushik, Raghu R, Maheshwar Reddy, Ankita Prasad, Sai Prasanna M S, "Ad analysis using Machine learning", PESIT – Bangalore South Campus.

[2] Anitha anandhan, Liyana Shuba, Maizatul Akmar Ismail and Ghulam Mujtaba "Social Media Recommender System: Review and Open Research Issues", IEEE Transaction on knowledge and data Engineering, date of publication February 27,2018.

[3] Ashutosh Bansal, Saleena B, Prakash B, "Using Data mining techniques to analyze the customer reaction toward social media."

[4] Bengio, Samy; Goodfellow, Ian J.; Kurakin, Alexey (2017)," Adversial Learning at Scala" Google AI.