

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

**A**

**Major Project Mid-term Report**

**on**

**PERSONALITY BASED MUSIC RECOMMENDATION SYSTEM**

Submitted By:

**Abhishek Paudel (070 BCT 502)**

**Brihat Ratna Bajracharya (070 BCT 513)**

**Miran Ghimire (070 BCT 521)**

**Nabin Bhattarai (070 BCT 522)**

Supervised By:

**Er. Daya Sagar Baral**

**Department of Electronics and Computer Engineering**

**IOE Central Campus, Pulchowk**

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING

IOE CENTRAL CAMPUS PULCHOWK

July 14, 2017

# ACKNOWLEDGEMENT

We would like to express our sincere gratitude to the Department of Electronics and Computer Engineering of IOE, Pulchowk Campus for providing us opportunity to implement the knowledge gained over these years as the Major Project for fourth year. Besides, we are highly indebted to our supervisor Mr. Daya Sagar Baral for providing valuable suggestion, guidance and motivation for this project. Finally we would also like to thank all of our friends who have directly and indirectly assisted us for selecting this project.

# ABSTRACT

With the evolution of internet and popularity of social media, it has become the most prominent and dominant source of information sharing and communication. A person's preference can be determined using his/her social media. Hence based on this information about his/her profile related content can be delivered to the user. Therefore we have come with Personality Based Music Recommendation System. This project mainly concentrates on the recommendation of the music based on the preference of the user which basically consists of the three parts as input,process and output. In input part, the system takes user data fetched from the social media and information about the music.In the processing media, the userdata and music data are analyzed for the recommender system (content based and collaborative filtering) along with the user's personality using Big 5 Personality Traits. Similarly, in the output part, it recommends the music to the user.

# TABLE OF CONTENTS

[ACKNOWLEDGEMENT ii](#_Toc487804637)

[ABSTRACT iii](#_Toc487804638)

[TABLE OF CONTENTS iv](#_Toc487804639)

[LIST OF FIGURES vii](#_Toc487804640)

[LIST OF TABLES viii](#_Toc487804641)

[LIST OF ABBREVIATIONS ix](#_Toc487804642)

[1. INTRODUCTION 1](#_Toc487804643)

[2. SUMMARY OF PROCESS 2](#_Toc487804644)

[2.1. Task Analysis 2](#_Toc487804645)

[2.2. Task Evaluation 3](#_Toc487804646)

[2.2.1. Task Completed/Running 3](#_Toc487804647)

[2.2.2. Task Remaining 3](#_Toc487804648)

[2.3. Schedule Analysis 4](#_Toc487804649)

[3. LITERATURE REVIEW 5](#_Toc487804650)

[3.1. Previous Work 5](#_Toc487804651)

[3.2. Preprocessing Text 6](#_Toc487804652)

[3.2.1. Tokenization 6](#_Toc487804653)

[3.2.2. Stemming 6](#_Toc487804654)

[3.2.3. Stop-words removal 6](#_Toc487804655)

[3.2.4. Term Weighting using TF-IDF 7](#_Toc487804656)

[3.3. Natural Language Processing 7](#_Toc487804657)

[3.3.1. Unigram model 8](#_Toc487804658)

[3.3.2. N - gram model 9](#_Toc487804659)

[3.3.3. Bag of Words Model 10](#_Toc487804660)

[3.4. Naive Bayes Classifier 11](#_Toc487804661)

[3.4.1. Gaussian Naive Bayes 11](#_Toc487804662)

[3.4.2. Multinomial Naive Bayes 12](#_Toc487804663)

[3.4.3. Bernoulli Naive Bayes 13](#_Toc487804664)

[3.4.4. Semi-supervised parameter estimation 14](#_Toc487804665)

[3.5. Recommender System 14](#_Toc487804666)

[3.5.1. Content-based filtering 16](#_Toc487804667)

[3.5.2. Collaborative filtering 16](#_Toc487804668)

[3.5.2.1. Memory based 17](#_Toc487804669)

[3.5.2.2. Model based approach 18](#_Toc487804670)

[3.5.3. Hybrid method 19](#_Toc487804671)

[4. METHODOLOGY 20](#_Toc487804672)

[4.1. Personality Dataset Analysis 20](#_Toc487804673)

[4.2. Feature Extraction from Personality DataSet 21](#_Toc487804674)

[4.3. Personality Analysis Model 21](#_Toc487804675)

[4.3.1. Naive Bayes Classifier 21](#_Toc487804676)

[4.3.1.1. Additive Smoothing 22](#_Toc487804677)

[4.3.1.2. Overfitting 23](#_Toc487804678)

[4.3.1.3. Underfitting 23](#_Toc487804679)

[4.3.1.4. Optimization 23](#_Toc487804680)

[4.4. Current Output Screenshots 24](#_Toc487804681)

[4.4.1. Stopwords used in the project 24](#_Toc487804682)

[4.4.2. Segment of Hash Table 24](#_Toc487804683)

[4.4.3. Accuracy of Naive-bayes Classifier 25](#_Toc487804684)

[4.4.4. UI Screenshots 25](#_Toc487804685)

[5. TOOLS AND TECHNOLOGIES USED 28](#_Toc487804686)

[5.1. Python Programming Language 28](#_Toc487804687)

[5.2. Django Web Development Framework 28](#_Toc487804688)

[5.3. Facebook Graph API 29](#_Toc487804689)

[6. PROBLEMS AND ISSUES 31](#_Toc487804690)

[6.1. Current Problems 31](#_Toc487804691)

[6.2. Issues 31](#_Toc487804692)

[6.3. Possible way out 31](#_Toc487804693)

[7. REFERENCES 32](#_Toc487804694)

# LIST OF FIGURES

2.1. Block Diagram of a System 2

2.3. PERT Activity of Node Diagram of Project 4

3.5. Various recommender system techniques 15

3.5.2. Collaborative filtering technique 17

4.1. AUTHID, STATUS and five personality traits 21

4.4.1. Stopwords 24

4.4.2. Segment of hash table 24

4.4.3. Accuracy and classifier output 25

4.4.4.1. Home Page 26

4.4.4.2. Facebook login dialog 26

4.4.4.3. Result showing personality of a Facebook user 27

# LIST OF TABLES

2.1. Project Schedule 3

# LIST OF ABBREVIATIONS

|  |  |
| --- | --- |
| **API** | Application Programming Interface |
| **AUTHID** | Authentication Identity |
| **CBF** | Content Based Filtering |
| **CF** | Collaborative Filtering |
| **DSF** | Django Software Foundation |
| **HTTP** | Hyper Text Transfer Protocol |
| **IR** | Information Retrieval |
| **MVC** | Model View Controller |
| **NB** | Naïve Bayes |
| **NLP** | Natural Language Processing |
| **NLTK** | Natural Language Tool Kit |
| **SVM** | Support Vector Machine |
| **TF-IDF** | Term Frequency - Inverse Document Frequency |
| **UI** | User Interface |
| **URL** | Uniform Resource Locator |

# 1. INTRODUCTION

“Personality Based Music Recommendation” is the system which uses social media profile of a person to recommend the appropriate music to that user. In this contemporary era of digital technologies, social media has become one of the prominent means for information sharing and communication. Likewise music has been one of the prominent market of entertainment. People listen to music everyday. The fact that music can blend with any emotion has made it’s way to different sorts of people with different sorts of personality.

Hence we have come up with the system to recommend the music to a different people based on their social media profiles. Previous work has shown that the information in users social media profiles is reflective of their actual personalities, not an idealized version of themselves, which makes social media platform for studying a people personality.

Several well studied personality models have been proposed, among which the Big Five model is established as the most popular one,which suggests that the regularity in someone’s behavior over time and situations uniquely identifies his/her personality type along five dimensions: Openness, Neuroticism, Extroversion, Agreeableness and Conscientiousness.

# 2. SUMMARY OF PROCESS

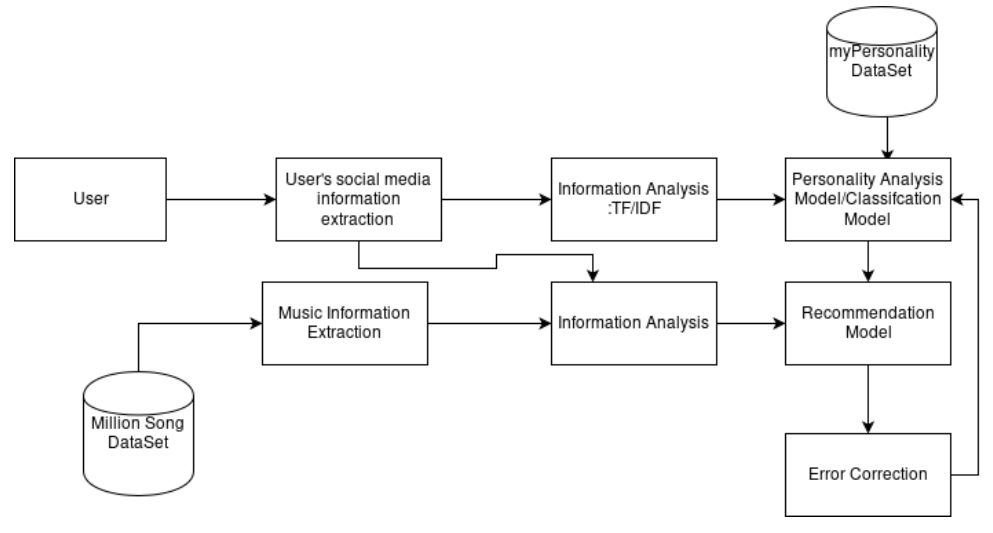


Figure 2.1. Block Diagram of a System

## 2.1. Task Analysis

|  |  |  |  |
| --- | --- | --- | --- |
| Activity | Description | Immediate Predecessors | Time (in weeks) |
| A | Personality Dataset Analysis | - | 1 |
| B | Feature Extraction from Personality Dataset | A | 3 |
| C | Personality Analysis Model | B | 4 |
| D | User Social Media Information Extraction | - | 1 |
| E | User Personality Analysis | C,D | 2 |
| F | Music Dataset Analysis | - | 1 |
| G | Feature Extraction from Music Dataset | F | 2 |
| H | Recommendation Model | E,G | 4 |
| I | Testing and Debugging | H | 2 |

Table 2.1. Project Schedule

## 2.2. Task Evaluation

### 2.2.1. Task Completed/Running

1. Personality DataSet Analysis
2. Feature Extraction from Personality DataSet
3. Personality Analysis Model
4. User Social Media Information Extraction
5. User Personality Analysis
6. Music DataSet Analysis
7. Feature Extraction from Music DataSet

### 2.2.2. Task Remaining

1. Recommendation Model
2. Testing and Debugging

## 2.3. Schedule Analysis

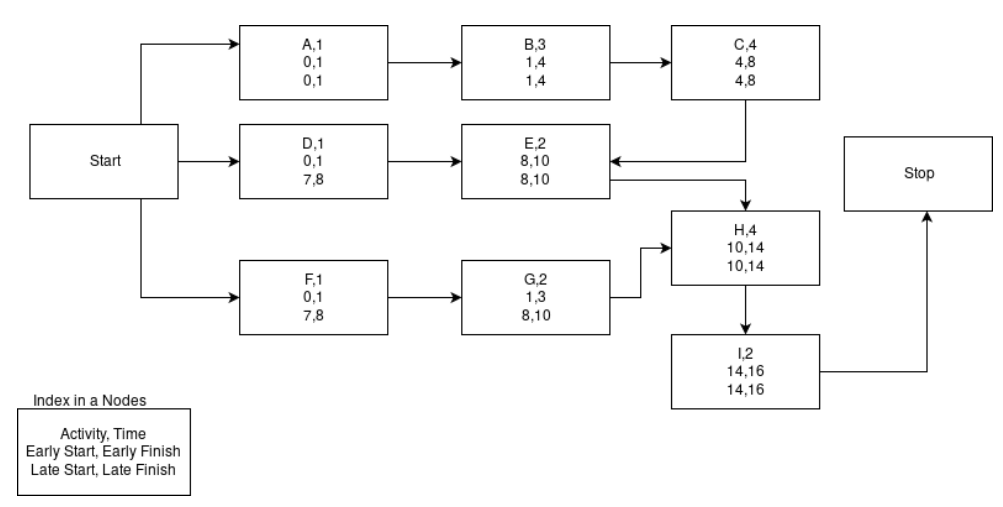


Figure 2.3. PERT Activity of Node Diagram of Project

# 3. LITERATURE REVIEW

Data mining is used for finding useful information from large amount of data. Some research related areas that uses data mining techniques includes text mining, web mining, image mining, sequential pattern mining, structure mining, graph mining and many others. Among these areas, we are particularly interested in text mining that deals with extracting information from textual data. Text mining techniques can easily handle semi-structured data and unstructured data apart from structured data documents.

## 3.1. Previous Work

There has been several attempts to enhance the performance of the music recommendation engine with the help of personality. Few of the notable works are:

1. "Improving music recommender systems: What can we learn from research on music tastes?" ISMIR published by 2014[12]. It suggests how a music recommendation can be enhanced.
2. "A Comparative Analysis of Personality-Based Music Recommender Systems." EMPIRE @ RecSys published in 2016[10]. It attempts to find the similar user with the help of the personality in collaborative filtering.
3. "Enhancing Music Recommender Systems with Personality Information and Emotional States: A Proposal." UMAP Workshops, presented in 2014[11]. It attempts to maps the personality of the user with the emotional state of the music.

In our project the work is being done to enhance the model built on second work paper listed above by designing the weighted hybrid system.

## 3.2. Preprocessing Text

Computerization of retrieval task and sequence of actions should be taken first to preprocess documents using natural language processing techniques. Usually, this sequence comprises tokenization, stemming and stop words removal.

### 3.2.1. Tokenization

A word token is a maximum sequence of consecutive characters without any delimiters. A delimiter can be a space, punctuation mark like comma or else. Tokenization is the process of parsing a character stream into sequence of word tokens by splitting the stream by the delimiters.

### 3.2.2. Stemming

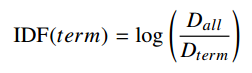
After tokenization, we need to reduce the token words into their ground form. This is important because in English or any other languages, a single word has variety of grammatical forms. Stemming facilitates this process by reducing a word to its stem or root form by simply removing its ending. Another process called Lemmatization allows us to reduce various linguistic forms of a word to their common canonical form. The primary objective is to replace synonymous words with a basic native form.

### 3.2.3. Stop-words removal

Last technique of pre-processing is stop-words removal. Stop-words are those words carrying little helpful information for information retrieval task. These include pronouns such as “it”, “he” and “she” link verbs such as “is”, “am” and “are”, etc. In addition to the list we have added own stop words like “you.”, “you…”, “you've”, etc.

### 3.2.4. Term Weighting using TF-IDF

TF-IDF (Term Frequency, Inverse Document Frequency) is a common term weighting scheme. It is a statistical approach to evaluating the importance of term in a corpus. TF is a local importance measure. Given a term and a document, in general, TF corresponds to the number of times the term appears within the document. Different from TF, IDF is a global importance measure most commonly calculated by the formula within the corpus,



where,

*Dall* = number of documents in the corpus

*Dterm* = number of documents containing *term*

Given a term, the fewer documents it is contained in, the more important it becomes. In our approach, we employ both TF and IDF to weigh terms.

## 3.3. Natural Language Processing

Natural language processing (NLP) is a field of computer science, artificial intelligence and computational linguistics concerned with the interactions between computers and human (natural) languages, and, in particular, concerned with programming computers to fruitfully process large natural language corpora. Challenges in natural language processing frequently involve natural language understanding, natural language generation (frequently from formal, machine-readable logical forms), connecting language and machine perception, dialog systems, or some combination thereof.

A statistical language model is a probability distribution over sequences of words. Given such a sequence, say of length n, it assigns a probability to the whole sequence. Having a way to estimate the relative likelihood of different phrases is useful in many natural language processing applications, especially ones that generate text as an output. Language modeling is used in speech recognition, machine translation, part-of-speech (POS) tagging, parsing, handwriting recognition, information retrieval and other applications.

Language models are used in information retrieval in the query likelihood model. Here a separate language model is associated with each document in a collection. Documents are ranked based on the probability of the query Q in the document's language model. Commonly, the unigram language model is used for this purpose—otherwise known as the bag of words model. Data sparsity is a major problem in building language models. Most possible word sequences will not be observed in training. One solution is to make the assumption that the probability of a word only depends on the previous n words. This is known as an n-gram model or unigram model when n = 1.

### 3.3.1. Unigram model

A unigram model used in information retrieval can be treated as the combination of several one-state finite automata. It splits the probabilities of different terms in a context, e.g. from

unigram.PNG

In this model, the probability of each word only depends on that word's own probability in the document, so we only have one-state finite automata as units. The automaton itself has a probability distribution over the entire vocabulary of the model, summing to 1.

### 3.3.2. N - gram model

In an n-gram model, the probability of observing the sentence is approximated as

ngram.PNG

Here, it is assumed that the probability of observing the ith word wi in the context history of the preceding i − 1 words can be approximated by the probability of observing it in the shortened context history of the preceding n − 1 words (nth order Markov property).

The conditional probability can be calculated from n-gram model frequency counts:

ngram2.PNG

The words bigram and trigram language model denote n-gram model language models with n = 2 and n = 3, respectively.

Typically, however, the n-gram model probabilities are not derived directly from the frequency counts, because models derived this way have severe problems when confronted with any n-grams that have not explicitly been seen before. Instead, some form of smoothing is necessary, assigning some of the total probability mass to unseen words or n-grams. Various methods are used, from simple "add-one" smoothing (assign a count of 1 to unseen n-grams) to more sophisticated models, such as Good-Turing discounting or back-off models.

### 3.3.3. Bag of Words Model

The bag-of-words model is a simplifying representation used in natural language processing and information retrieval (IR). 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. The bag-of-words model is commonly used in methods of document classification where the (frequency of) occurrence of each word is used as a feature for training a classifier. The Bag of words consist of two models:

1. Continuous-Gram Model: In this model in which every single words that appears in the document is treated as the feature in the sequence in which they appear. In this model both the presence of the word and appearance sequence matters.
2. Skip-Gram Model: It is the model in which every single words that appears in the document is treated as the feature. However the sequence in which they appear doesn’t matter.

The bag of words model is the most commonly used methods in sentence classification where the frequency of occurrence of each word is used as a feature for training a classifier. After the transforming the text into the bag of words, we calculate various measures to characterize the text. The most common type of characteristics calculated from the bag of word is term-frequency i.e. the number of times the term appears in the text. In case, of our project, skip-gram model has been used because the existence of the word in a status posted by the user matters not the sequence in which they appear. After the bag of words extraction, term-frequency has been calculated for the classification purpose.

## 3.4. Naive Bayes Classifier

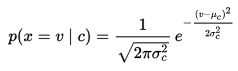
In machine learning, naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Naive Bayes classifiers are highly scalable, requiring a number of parameters linear in the number of variables (features/predictors) in a learning problem. Maximum-likelihood training can be done by evaluating a closed-form expression, which takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

In the statistics and computer science literature, Naive Bayes models are known under a variety of names, including simple Bayes and independence Bayes. All these names reference the use of Bayes' theorem in the classifier's decision rule, but naive Bayes is not (necessarily) a Bayesian method. Some variant of naive bayes are briefed below.

### 3.4.1. Gaussian Naive Bayes

When dealing with continuous data, a typical assumption is that the continuous values associated with each class are distributed according to a Gaussian distribution. For example, suppose the training data contains a continuous attribute, x. We first segment the data by the class, and then compute the mean and variance of x in each class. Let be the mean of the values in x associated with class *c*, and let be the variance of the values in x associated with class *c*. Suppose we have collected some observation value v. Then, the probability *distribution* of v given a class c, , can be computed by plugging v into the equation for a Normal distribution parameterized by and . That is,



### 3.4.2. Multinomial Naive Bayes

Naive Bayes is a learning algorithm employed frequently to tackle text classification problems because of its efficiency and easiness to implement. There are other two models that are commonly used: the multivariate Bernoulli event model and the multinomial event model. The latter outperforms the multivariate one though it is still inferior to support vector machine (SVM). lets now discuss how MNB computes class probabilities for a given document.

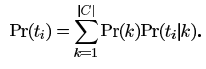
Let the set of classes be denoted by C and N be the size of our vocabulary. Then using bayes rule,



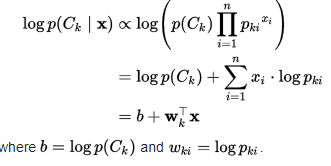
The class prior Pr(c) is estimated by dividing the number of documents belonging to class c by the total number of documents. Probability of obtaining a document like in class c is



Where fni is the count of word n in our test document ti and is the probability of word n given class c. Also the normalizing factor can be computed as



The multinomial naive Bayes classifier becomes a linear classifier when expressed in log-space:



If a given class and feature value never occur together in the training data, then the frequency-based probability estimate will be zero. This is problematic because it will wipe out all information in the other probabilities when they are multiplied. Therefore, it is often desirable to incorporate a small-sample correction, called pseudocount, in all probability estimates such that no probability is ever set to be exactly zero. This way of regularizing naive Bayes is called Laplace smoothing when the pseudocount is one, and Lidstone smoothing in the general case.

### 3.4.3. Bernoulli Naive Bayes

In the multivariate Bernoulli event model, features are independent boolean i.e. binary variables that describe inputs. This model is popular for document classification tasks, where binary term occurrence features are used rather than term frequencies. If is a boolean expressing the occurrence or absence of the *i*th term from the vocabulary, then the likelihood of a document given a class is given by

bernoulli.PNG

where is the probability of class generating the term . This event model is especially popular for classifying short texts. It has the benefit of explicitly modelling the absence of terms.

### 3.4.4. Semi-supervised parameter estimation

Given a way to train a naive Bayes classifier from labeled data, it's possible to construct a semi-supervised training algorithm that can learn from a combination of labeled and unlabeled data by running the supervised learning algorithm in a loop.

Given a collection of labeled samples L and unlabeled samples U, start by training a naive Bayes classifier on L.

Until convergence, do:

Predict class probabilities for all examples x in D.

Re-train the model based on the probabilities (not the labels) predicted in the previous step. Convergence is determined based on improvement to the model likelihood , where denotes the parameters of the naive Bayes model.

This training algorithm is an instance of the more general expectation–maximization algorithm (EM): the prediction step inside the loop is the *E*-step of EM, while the re-training of naive Bayes is the *M*-step. The algorithm is formally justified by the assumption that the data are generated by a mixture model, and the components of this mixture model are exactly the classes of the classification problem.

## 3.5. Recommender System

Recommender systems are information filtering systems that deal with the problem of information overload by filtering vital information fragment out of large amount of dynamically generated information according to user’s preferences, interest, or observed behavior about item. Recommender system has the ability to predict whether a particular user would prefer an item or not based on the user’s profile.

Recommender systems are beneficial to both service providers and users. Recommendation systems have also proved to improve decision making process and quality. In e-commerce setting, recommender systems enhance revenues, for the fact that they are effective means of selling more products. In scientific libraries, recommender systems support users by allowing them to move beyond catalog searches. Therefore, the need to use efficient and accurate recommendation techniques within a system that will provide relevant and dependable recommendations for users cannot be over-emphasized.



Figure 3.5. Various recommender system techniques

Recommender systems typically produce a list of recommendations in one of two ways – through collaborative and content-based filtering or the personality-based approach.Collaborative filtering approaches build a model from a user's past behaviour (items previously purchased or selected and/or numerical ratings given to those items) as well as similar decisions made by other users. This model is then used to predict items (or ratings for items) that the user may have an interest in. Content-based filtering approaches utilize a series of discrete characteristics of an item in order to recommend additional items with similar properties. These approaches are often combined to form Hybrid Recommender Systems.

### 3.5.1. Content-based filtering

Content-based technique is a domain-dependent algorithm and it emphasizes more on the analysis of the attributes of items in order to generate predictions. When documents such as web pages, publications and news are to be recommended, content-based filtering technique is the most successful. In content-based filtering technique, recommendation is made based on the user profiles using features extracted from the content of the items the user has evaluated in the past. Items that are mostly related to the positively rated items are recommended to the user. CBF uses different types of models to find similarity between documents in order to generate meaningful recommendations. It could use Vector Space Model such as Term Frequency Inverse Document Frequency (TF/IDF) or Probabilistic models such as Naïve Bayes Classifier, Decision Trees or Neural Networks to model the relationship between different documents within a corpus. These techniques make recommendations by learning the underlying model with either statistical analysis or machine learning techniques. Content-based filtering technique does not need the profile of other users since they do not influence recommendation. Also, if the user profile changes, CBF technique still has the potential to adjust its recommendations within a very short period of time. The major disadvantage of this technique is the need to have an in-depth knowledge and description of the features of the items in the profile.

### 3.5.2. Collaborative filtering

Collaborative filtering is a domain-independent prediction technique used for content that cannot easily and adequately be described by metadata such as movies and music. Collaborative filtering technique works by building a database (user-item matrix) of preferences for items by users. It then matches users with relevant interest and preferences by calculating similarities between their profiles to make recommendations.

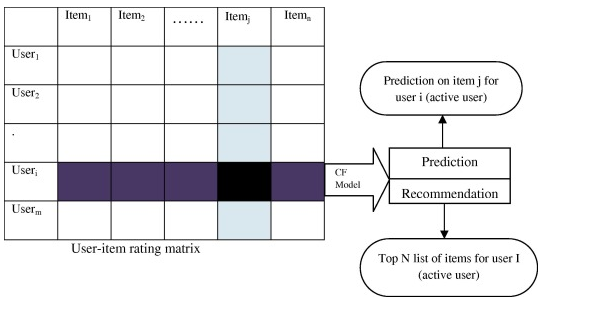


Figure 3.5.2. Collaborative filtering technique

Such users build a group called neighborhood. An user gets recommendations to those items that he has not rated before but that were already positively rated by users in his neighborhood. Recommendations that are produced by CF can be of either prediction or recommendation. Prediction is a numerical value, Rij, expressing the predicted score of item j for the user i, while Recommendation is a list of top N items that the user will like the most as shown in figure below. The technique of collaborative filtering can be divided into two categories: memory-based and model-based.

#### 3.5.2.1. Memory based

This approach uses user rating data to compute the similarity between users or items. This is used for making recommendations. This was an early approach used in many commercial systems. It's effective and easy to implement. Typical examples of this approach are neighbourhood-based CF and item-based/user-based top-N recommendations. The user based top-N recommendation algorithm uses a similarity-based vector model to identify the k most similar users to an active user. After the k most similar users are found, their corresponding user-item matrices are aggregated to identify the set of items to be recommended.

The advantages with this approach include: the explainability of the results, which is an important aspect of recommendation systems; easy creation and use; easy facilitation of new data; content-independence of the items being recommended; good scaling with co-rated items.There are also several disadvantages with this approach. Its performance decreases when data gets sparse, which occurs frequently with web-related items. This hinders the scalability of this approach and creates problems with large datasets. Although it can efficiently handle new users because it relies on a data structure, adding new items becomes more complicated since that representation usually relies on a specific vector space. Adding new items requires inclusion of the new item and the re-insertion of all the elements in the structure.

#### 3.5.2.2. Model based approach

In this approach, models are developed using different data mining, machine learning algorithms to predict users' rating of unrated items. There are many model-based CF algorithms. Bayesian networks, clustering models, latent semantic models such as singular value decomposition, probabilistic latent semantic analysis, multiple multiplicative factor, latent Dirichlet allocation and Markov decision process based models.

In this model, methods like singular value decomposition, principle component analysis, known as latent factor models, compress user-item matrix into a low-dimensional representation in terms of latent factors. One advantage of using this approach is that instead of having a high dimensional matrix containing abundant number of missing values we will be dealing with a much smaller matrix in lower-dimensional space. A reduced presentation could be utilized for either user-based or item-based neighborhood algorithms. It handles the sparsity of the original matrix better than memory based ones. Also comparing similarity on the resulting matrix is much more scalable especially in dealing with large sparse datasets

### 3.5.3. Hybrid method

Hybrid filtering technique combines different recommendation techniques in order to gain better system optimization to avoid some limitations and problems of pure recommendation systems. Combining of algorithms will provide more accurate and effective recommendations than a single algorithm as the disadvantages of one algorithm can be overcome by another algorithm. Using multiple recommendation techniques can suppress the weaknesses of an individual technique in a combined model. The combination of approaches can be done in any of the following ways:

1. separate implementation of algorithms and combining the result
2. utilizing some content-based filtering in collaborative approach
3. utilizing some collaborative filtering in content-based approach
4. creating a unified recommendation system that brings together both approaches.

# 4. METHODOLOGY

## 4.1. Personality Dataset Analysis

In order to predict a personality of social media users, myPersonality dataset was used. It consists of

1. AUTHID: User\_Id, it was represented with a unique random number in order to protect user’s identity.
2. STATUS: It consisted of status posted by the user’s in certain time period frame. It consisted of total of 9918 status post of more than 250 users.
3. PERSONALITY CLASSIFICATION: It classifies personality as Big Five Personality traits. These traits are:
4. Openness to Experience: curious, intelligent, imaginative. High scorers tend to be artistic and sophisticated in taste and appreciate diverse views, ideas and experiences.
5. Conscientiousness: responsible, organized, persevering. Conscientious individuals are extremely reliable and tend to be high achievers, hard workers and planners.
6. Extroversion: outgoing, amicable, assertive. Friendly and energetic, extroverts, draw inspiration from social situations.
7. Agreeableness: cooperative, helpful, nurturing. People who score high in agreeableness are peace-keepers who are generally optimistic and trusting of others.
8. Neuroticism: anxious, insecure, sensitive. Neurotics are moody, tense and easily tipped into experiencing negative emotions.

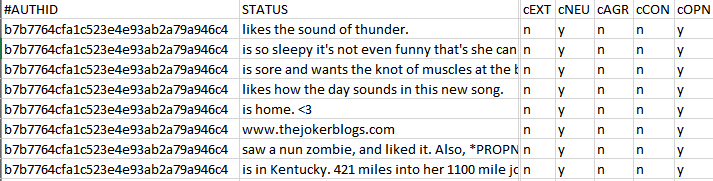


Figure 4.1. AUTHID, STATUS and five personality traits

## 4.2. Feature Extraction from Personality DataSet

Required features from myPersonality Dataset was extracted by performing series of process of tokenization, stemming and stop-words removal.

## 4.3. Personality Analysis Model

For creating model for personality analysis based on the status update in social media, Naive Bayes Classifier was used.

### 4.3.1. Naive Bayes Classifier

It is one of the model used for the classification under the “Bayesian Classifier”. In machine learning, Naive Bayes classifier are family of simple probabilistic classifiers based on Bayes’s theory with assumption of “independence” between the features. If the dependence between the features exists Bayesian Network will be used for the classification. The major advantage of the Naive Bayes is it’s simplicity and highly scalable feature and ability to work easily on huge dataset. Despite the oversimplified assumptions, Naive Bayes have worked quite well in many complex real-world situations. For problem formulation of Naive Bayes, for given the set of features , Naive Bayes classifies into a class c as:

where, P(Ck) is known as prior probability and, P(x|Ck) is known as posterior probability. In practice, only the numerator in above fraction is taken into account since the denominator does not depend on C and the values of the features for a given set of features are constant, so the denominator is effectively constant. Hence in order to maximize the function denominator is ignored and the model becomes:

which can also be written as:

P(Ck|x) = prior\_probability ∗ posterior\_probability

For a classification of sentence or document “Multinomial Naive Bayes” is one of the popular model which has been used in the project. With a multinomial model, features represent the frequencies with which certain events have been generated by multinomial i.e feature set.

#### 4.3.1.1. Additive Smoothing

In statistics, additive smoothing, also called Laplace smoothing, is a technique used to smooth categorial data. Given an observation from a multinomial distribution with N trials and parameter vector , smoothed version of the data gives the estimator

when α = 1, it’s called Add-One Laplace smoothing. This technique is used in this project.

#### 4.3.1.2. Overfitting

In order to reduce the overfitting of the classifier, kth-fold cross validation technique is used. The advantage of this method is that all observations are used for both training and validation, and each observation is used for validation exactly once. In this project, 5th-fold cross validation technique is applied in which the data set is divided into the 5 test cases and train cases and classifier is trained on each of those cases.

#### 4.3.1.3. Underfitting

Underfitting in the Naive Bayes classifier, can occur whenever the probabilities, prior and/or posterior are very small. In such case, in order to prevent the program from underfitting resulting from the multiplication of the very small terms, log of probability is taken after which the resulting equation becomes:

#### 4.3.1.4. Optimization

Naive Bayes Classifier as seen from the above equation, classifies feature set into a class via the multiplication of the prior and posterior probability which requires their computation each time the classifier tries to classify the feature set into class.

In order to solve the above problem, posterior probability is precomputed and stored on Hash Table, where it’s stores the posterior probability of each feature set, which can be easily be retrieved and used for the classification in the program.

## 4.4. Current Output Screenshots

### 4.4.1. Stopwords used in the project

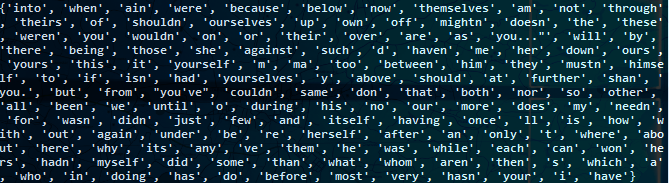


Figure 4.4.1. Stopwords

### 4.4.2. Segment of Hash Table

The screenshot shown here is a segment of hash table consisting of 1409 rows of and five column of personality traits. For each word, traits values corresponding to Agreeableness, Conscientiousness, Extraversion, Neuroticism, and Openness respectively are shown.

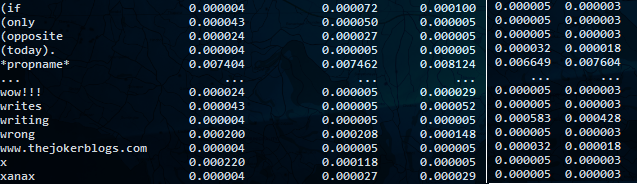


Figure 4.4.2. Segment of hash table

### 4.4.3. Accuracy of Naive-bayes Classifier

Output screenshot below shows the accuracy calculated via five fold cross validation of naive bayes classifier. This output also shows the result of classification done via this classifier on a given string.

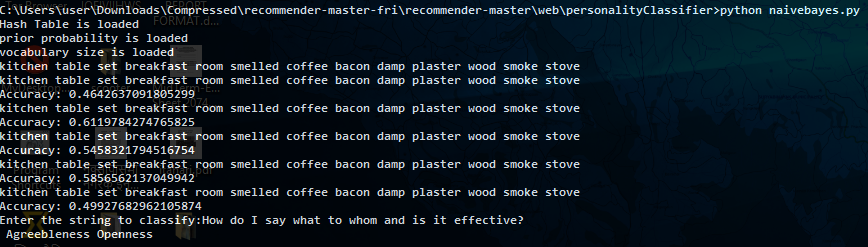


Figure 4.4.3. Accuracy and classifier output

### 4.4.4. UI Screenshots

This part consists of some UI screenshots.

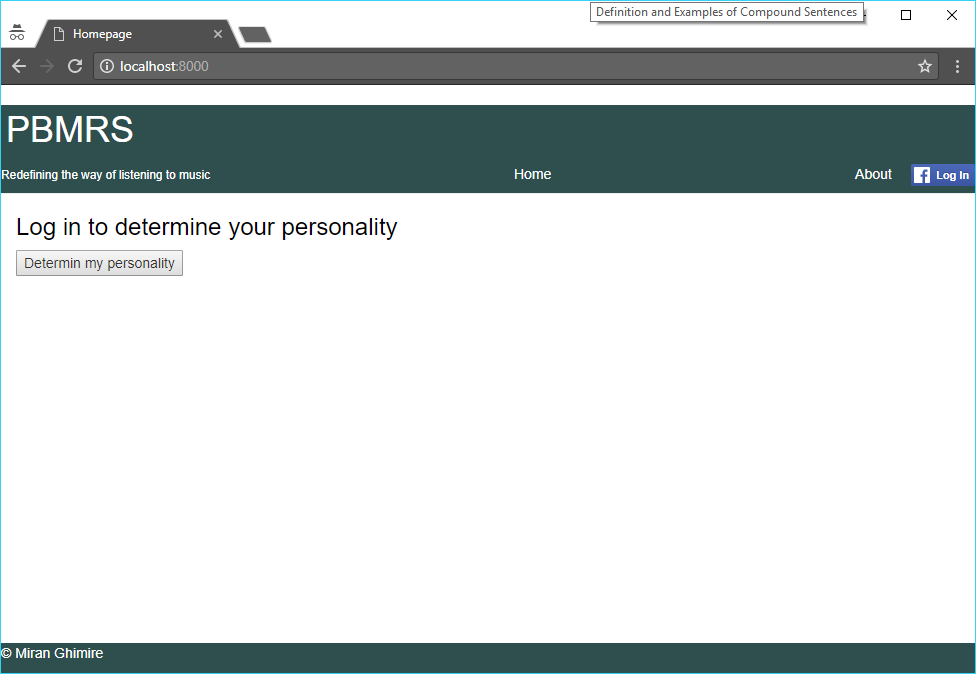


Figure 4.4.4.1. Home Page

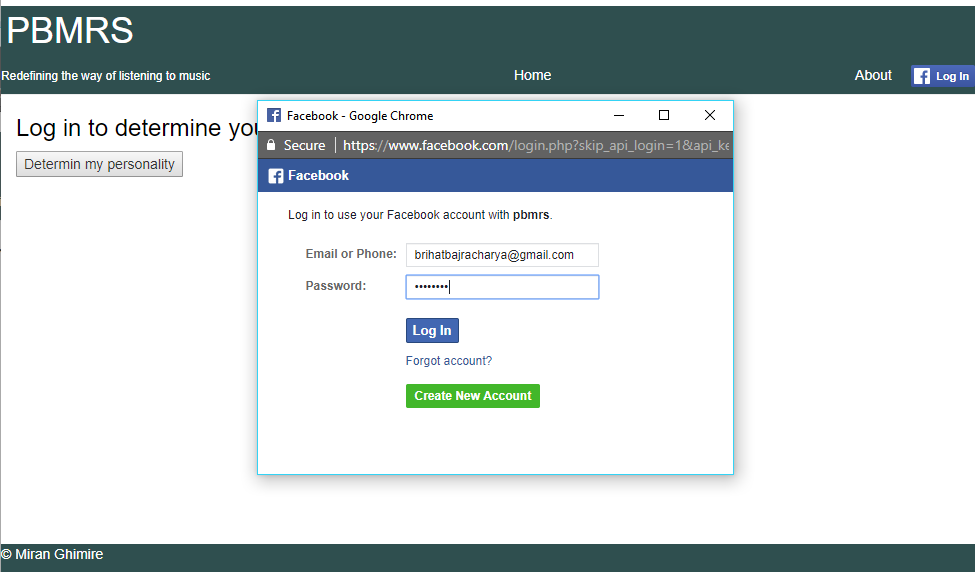


Figure 4.4.4.2. Facebook login dialog

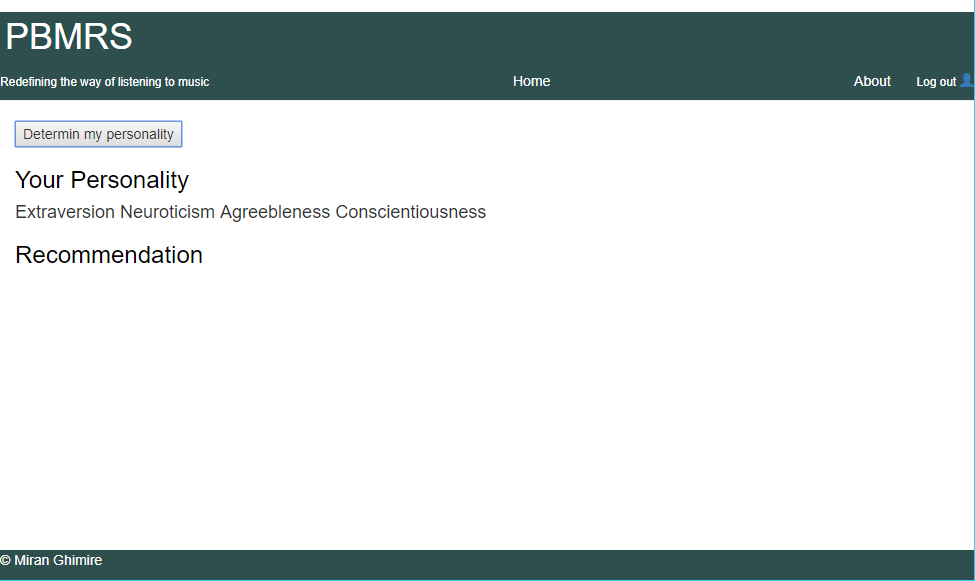


Figure 4.4.4.3. Result showing personality of a Facebook user

# 5. TOOLS AND TECHNOLOGIES USED

Python Programming Language was used for overall logical description of the project and Django is employed to make UI. Python is very popular programming language for data science. Below is a brief description about Python Programming Language and Django Web Development Framework

## 5.1. Python Programming Language

Python is a widely used high-level, general-purpose, interpreted, dynamic programming language. Its design philosophy emphasizes code readability, and its syntax allows programmers to express concepts in fewer lines of code than possible in languages such as C++ or Java. The language provides constructs intended to enable clear programs on both a small and large scale. Python supports multiple programming paradigms, including object oriented, imperative and functional programming or procedural styles. It features a dynamic type system and automatic memory management and has a large and comprehensive standard library.

Python interpreters are available for many operating systems, allowing Python code to run on a wide variety of systems. Using third-party tools, such as Py2exe or Pyinstaller, Python code can be packaged into stand-alone executable programs for some of the most popular operating systems, so Python-based software can be distributed to, and used on, those environments with no need to install a Python interpreter.

## 5.2. Django Web Development Framework

Django is a free and open-source web framework, written in Python, “for perfectionists with deadline”. It follows the model-view-controller (MVC) architectural pattern. It is maintained by the Django Software Foundation (DSF), an independent nonprofit organization. Django’s primary goal is to ease the creation of complex, database-driven websites. Django emphasizes re-usability and“pluggability” of components, rapid development, and the principle of don’t repeat yourself. Python is used throughout, even for settings files and data models. Django also provides an optional administrative create, read, update and delete interface that is generated dynamically through introspection and configured via admin models. Some well known sites that use Django include Pinterest, Instagram, Mozilla, The Washington Times, Disqus, the Public Broadcasting Service, BitBucket, and Nextdoor. Despite having its own nomenclature, such as naming the callable objects generating the HTTP responses “views”, the core Django framework can be seen as an MVC architecture. It consists of an object-relational mapper (ORM) that mediates between data models (defined as Python classes) and a relational database (“Model”), a system for processing HTTP requests with a web templating system (“View”), and a regular-expression-based URL dispatcher (“Controller”).

For developing a Django project, no special tools are necessary, since the source code can be edited with any conventional text editor. Nevertheless, editors specialized on computer programming can help increase the productivity of development, e.g. with features such as syntax highlighting. Since Django is written in Python, text editors which are aware of Python syntax are beneficial in this regard. Integrated development environments (IDE) add further functionality, such as debugging, refactoring, unit testing, etc. As with plain editors, IDEs with support for Python can be beneficial. Some IDEs that are specialized on Python additionally have integrated support for Django projects, so that using such an IDE when developing a Django project can help further increase productivity.

## 5.3. Facebook Graph API

The Graph API is the primary way to get data out of, and put data into Facebook's platform. It's a low-level HTTP-based API that you can use to programmatically query data, post new stories, manage ads, upload photos, and perform a variety of other tasks that an app might implement.

The Graph API is named after the idea of a 'social graph' - a representation of the information on Facebook composed of nodes (basically "things" such as a User, a Photo, a Page, a Comment), edges (the connections between those "things", such as a Page's Photos, or a Photo's Comments), and fields (info about those "things", such as a person's birthday, or the name of a Page). The Graph API is HTTP-based, so it works with any language that has an HTTP library, such as cURL and urllib. We can also use the Graph API directly in your browser, for example a Graph API request is equivalent to

GET graph.facebook.com/facebook/picture?redirect=false

# 6. PROBLEMS AND ISSUES

After training the personality classifier, the accuracy of the classifier was not so satisfactory.

## 6.1. Current Problems

Since the format of the status/post in social media is open, extraction of useful features from social media post has been a challenge.

## 6.2. Issues

1. Training takes time i.e. a bit time consuming.
2. Accuracy of the classifier is not satisfactory.

## 6.3. Possible way out

1. Stop Words scope can be increased via the introduction of the additional stop words which might help to increase the accuracy of the classifier.
2. N-gram model might also be useful in increasing the accuracy.
3. Different models such as SVM, logistic regression can be build as the performance test measures.
4. Training time can be reduced with the help of the optimization again where broadening the scope of the stop words can be very useful.

# 7. REFERENCES

1. myPersonality DataSet. http://mypersonality.org/wiki/doku.php
2. Kaggle Million Song Dataset. https://labrosa.ee.columbia.edu/millionsong
3. Christopher Manning, Stanford NLP - Stanford NLP Group. https://nlp.standford.edu/manning/
4. Recommendation System, Coursera. https://coursera.org/learn/machine-learning
5. Studying the big five personality traits-UK Essays. https://ukessays.com/essays/psychology/studying-the-big-five-personality-traits.php
6. Facebook Graph API. https://developers.facebook.com/docs/graph-api
7. A. M. Kibriya, E. Frank, B. Pfahringer, and G. Holmes. *Multinomial Naive Bayes for Text Categorization Revisited*. (2008)
8. Dr. S. Vijararani, J. Ilamathi, Nithya. Preproessing Techniques for Text Mining - An Overview, in *International Journal of Computer Science & Communication Networks,* Vol 5(1), 7-16
9. F. O. Isinkaye, Y. O. Folajimi, B. A. Ojokoh. *Recommendation systems: Principles, methods and evaluation.* http://www.sciencedirect.com/science/article/pii/S1110866515000341
10. Onori, Melissa, Alessandro Micarelli, and Giuseppe Sansonetti. "A Comparative Analysis of Personality-Based Music Recommender Systems." *EMPIRE @ RecSys*. 2016
11. Ferwerda, Bruce, and Markus Schedl. "Enhancing Music Recommender Systems with Personality Information and Emotional States: A Proposal." *UMAP Workshops*. 2014.
12. Laplante, Audrey. "Improving music recommender systems: What can we learn from research on music tastes?." *ISMIR*. 2014.

|  |
| --- |
|  |

1. Burke, Robin. "Hybrid recommender systems: Survey and experiments." *User modeling and user-adapted interaction* 12.4 (2002): 331-370.