Faculty of Informatics and Computer Science

Computer Science

Machine Learning for Extraction and Classification of Adverse Drug Events from social Media

By: Ahmed Hossam Ahmed Mohamed ElSabbagh

Supervised By

Associate Prof. Nahla Barakat

And Professor Khaled Nagaty

June 2019

Abstract

Adverse Drug Reactions/Events (ADR) are a harmful or unpleasant reaction caused by the use of medicinal product [1], these effects are studied through the science of Pharmacovigilance, the science and activities relating to the detection, assessment, understanding and prevention of these effects [2] [3]. The ADR caused by drugs after their release is considered a major health hazard, causing a large number of death, it is estimated that more then 6% of patients are hospitalized as a cause of serious ADRs, which is translated into more than 2 million patients, 0.32% of whom are fatalities or more than 100 thousands deaths in the US alone [4].

This project aims to research and find a good method to implement an Adverse Drug Reaction (ADR) detection program using machine learning based methods, with dataset collected from social media and/or medical forums.

The dataset will be extracted from specialized online forums using a crawler. The user data and some physical traits, like age, gender, weight, height and blood pressure will be extracted from the dataset using text mining techniques, pattern matching and sliding window. Using UMLS Metathesaurus medical concepts using the help of MetaMap, ADRs and diseases will be identified, and their presence or absence is as both features- as medical history- and classes for supervised learning to predict the possibility of being inflicted by an ADR, specifically or generally.

The dataset will be also tested with unsupervised association rule mining using word embedding and association rule mining, in order to find patterns and to find statistical relation between drugs and any ADR in question, and how it connect the user profile and medical history.

Acknowledgment

Deep and special thanks for my supervisor, Associate prof. Nahla Barakat, without whose guidance, suggestions and support, this project would never have been made.

Sincere thanks to prof. Khaled Nagaty, for his suggestions and encouragement in the making of this project.

Special thanks to my family, especially my brother Mohamed, for being a constant aid and guidance throughout my entire academic life.

The code for Apriori algorithm is partially taken from Usman Malik's online tutorial from stack abuse, which referred to in the Apriori section.

Table of Contents

Table of Figures

List of tables

Chapter 1: Introduction

1.1. Overview

Adverse Drug Reactions/Events (ADR) are a harmful or unpleasant reaction caused by the use of medicinal products [1], these effects are studied through the science of Pharmacovigilance, the science and activities relating to the detection, assessment, understanding and prevention of these effects [2] [3]. The ADR caused by drugs after their release is considered a major health hazard, causing a large number of death, it is estimated that more then 6% of patients are hospitalized as a cause of serious ADRs, which is translated into more than 2 million patients, 0.32% of whom are fatalities or more than 100 thousands deaths in the US alone [4].

There are many steps taken Through medical trials to find the ADRs, however some people may not have the same side effects as others and therefore not all ADRs that affect all different patients appear until phase IV trials -postmarketing trials- especially since most medical trials are concentrated on certain demographics, therefore are many uncertainties of the effect of the drugs on any given population [5] [6], and despite the many mediums offered by the FDA to report ADR (FAERS, MERP, MedWatch) [3], 90% of the ADR are in fact under-reported [7].

However, a new field of pharmacovigilance via social media has been introduced in recent years, as there are many disease support networks (DailyStrength and MedHelp), patient forums (AskAPatient) and miniblogs(Twitter) [8], where patients are involved in sharing their experiences with certain drugs, with many of them and their caregivers actively read these experiences [3]. Many data mining techniques were adapted to extract potential ADRs of drugs through text mining and machine learning, allowing many researchers not only a new way research ADRs, but also help them find new ADRs they had not known to have previously existed in some drugs or find earlier occurances than previously reported [3] [9].

1.2. Scope and objectives

This project aims to research and find a good method to implement an Adverse Drug Reaction (ADR) detection program using machine learning based methods, with dataset collected from social media and/or medical forums.

It also focuses on unsupervised learning with word embedding, to try to find the most relevant terms used in the dataset, in order to find patterns within the dataset that could help in identifying common ADRs.

It will also include a section using association rules, where the relations between ADRs, diseases and drugs will be found, this will identify the most common diseases and their potential cause.

Finally the main focus will be on supervised learning, testing three techniques, namely Random Forests, SVM and Naïve Bayes. To try to predict the potential drug causing medical issue, or predict the chances of an ADR occurring based on the user profile and their medical history.

Chapter 2: Related Work

2.1. Background

There has been a lot of work related concerning ADR datamining, the most prevalent methodologies include lexicon-based pattern mining [5] [6] [9] [10] and supervised machine learning [11] [12] [13] approaches [3], however there are also rare cases of hybrid systems [14] [15], and partially supervised implementations [16]. The evaluation of these techniques was done mainly using three metrics, F-Score, Recall, and Precision [17] [18] [19].

2.2. Literature Survey

2.2.1. Traditional ADR Detection Methods

Electronic post marketing pharmacovigilance has traditionally been applied through FDA Adverse Event Reporting System (FAERS) [20], MedWatch [21] and the Institute of Safe Medication Practices Medication Error Reporting System (MERP) [3]. These methods however are voluntary for the public and healthcare professionals, meaning that data could be missing or incomplete, and clinical narratives are limited to researchers affiliated with medical research centers [3]. However, due to the widespread use of social media, the wider knowledge and the convenience of its use, patients have been more comfortable sharing their experiences and looking for answers as shown in a survey by Pew Research Center [22]. This is why many researchers have been trying to find ways to mine social media for ADR information.

2.2.2. Scope of research

There has been two types of research, direct ADR research as surveyed by [3], which focuses on trying to discover a relation between a given drug and any ADRs, the second is drug-drug-interaction (DDI) as surveyed by [23], which focuses on the side effects of consuming several drugs in the same time. This project will focus on the first type of research.

2.2.3. Finding Datasets

The first obstacle regarding ADR research is to find the necessary data. The data used in all projects is comprised of social media posts with mentions of drug names, these posts may or may not contain mentions of ADR related to the drug name. The posts are arranged into a corpus of data which is used to train and evaluate machine learning and pattern mining approaches [3] [6] [9] [10] [11] [12] [13] [14] [16]. The social media in question are twitter, the data of which is available at Arizona State University [3] [11] [14] [24], and DailyStrength [25], MedHelp [26], PatientsLikeMe [27], Yahoo! Forums [16], Medications.Com [28] and AskAPatient [29]. Using either purpose built crawlers [5] or API [12] to get the data.

2.2.4. General Text Processing Techniques:

Generally, the following sequence of text processing techniques were used to make the data usable for learning process, with variations and different tools used to achieve it, the texts were split in the punctuations and whitespaces, creating tokens (Tokenization), stemming is applied to remove similar words or turn them to a simpler form (training, trainer, trained = train), stop words (a, and, but) are removed completely. The tokens are then turned into vectors, from which certain features are extracted and used in the learning algorithm [30].
Many researchers used lexicon for comparison, provided by UMLS Metathesaurus [31] , SIDER [32], MedEffect Canada [33] and manually allocated colloquial terms.

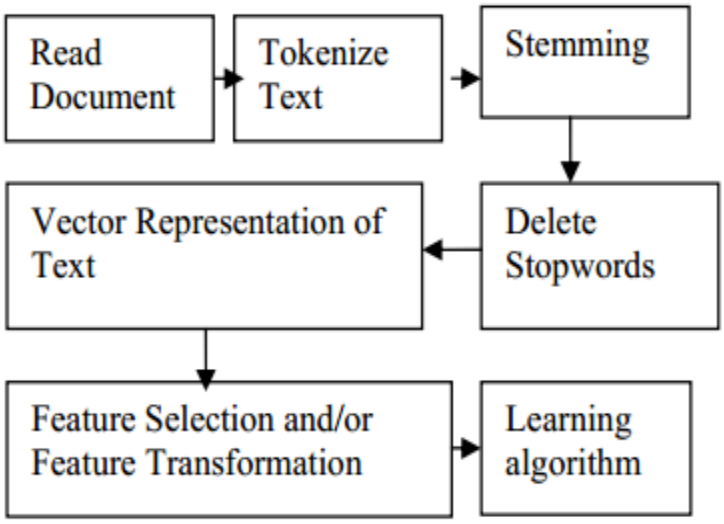


Figure 1 Text Classification Procedure [30]

2.2.5. Lexicon-Based Techniques

The first real attempt to create a pharmacovigilance program was in 2010 by Leaman et al. [5]. The lexicon is extracted from UMLS Metathesaurus [31] , SIDER [32], MedEffect Canada [33] and manually allocated colloquial terms [5]. It used a purpose built parallelized crawler to extract comments from DailyStrength [25], bringing about 3600 annotated comments and 450 reserved for evaluation, the information was taken for the following drugs, carbamazepine, olanzapine, trazodone, and ziprasidone [5]. The comments were annotated for adverse effects, beneficial effects, indications and other terms [5], where the following tools are using in text processing, Java (Tokenization), Snowball implementation (Stemming) [34], Jaro-Winkler measurement [35] (Similarity for misspelling) [5]. A basic lexical similarity comparison was applied, where a sliding window of tokens (size = 5) was go over the tokenized text, comparing them with lexical terms in the dictionary, pairing them in an assignment problem [5], the similarities were summed and normalized with the result by the number of tokens in lexical term [5]. The closest verbs were used to categorize the mention, where verbs like “taking” was an indication, since ADR are targeted, Indications, beneficial effects and others were filtered out [5]. 1260 adverse effects, 391 indications, 157 beneficial effects and 78 other, for a total of 1,886 annotations [5]. For the evaluation, Precision = 78.3%, recall = 69.9%, for an F-Score = 73.9%, not all known ADRs were recognized [5].

The following lexicon methods that followed use association rule, a data mining approach that tries to find statistical relation between drugs and ADRs, the goal is to find enough minimum support and confidence constraints [36]. The research by [6] [10] were among the first to apply this method.

The first research [6], which was made by the team working on [5], tried to match a comment with certain grammatical patterns in order to find the mention of ADR and detecting whether a person is actually inflicted or not [6]. To achieve that, the DailyStrength [25] dataset corpus was used, with an additional 3290 records were added for a total of 6890 records, including the original 1886 annotations from the previous research and using the same dictionary as before and the same drugs [6]. Then to solve the problem, it follows three steps: 1) Term Sequence Generation, the sequence of words in which an ADR is mentioned is stored in a file, each line in the file has the ADR replaced with ADR keyword, part of speech (POS) tagging is performed

using the Stanford parser [6] [37]. Some POS were kept (like verbs) using Wordnet [6] [38]. Thus a term is created. 2) Frequent Rule Identification, Apriori tool [39] (which implements Apriori Algorithm) is used to mine association rule, where a term like “make PRP RB CC =>ADR” (which is a combination of verb make and POS tags) occurs when ADR is reported [6]. 3) Frequent Pattern Generation, where patterns based on Frequent Rule Identification are generated, with short patterns and patterns with placeholders after ADR being excluded. Precision = 70.01%, recall = 66.32% for an F-Score = 67.96%, with minimum support = 4, maximum=6, minimum number of terms per rule=4, other tunings and replacements in the input were tried, but this was by far the best result in all the attempts [6].

While [10] tried to detect ADRs through data in parenting sites using disproportionality techniques using the concepts discussed in the three Dutch examples methods [10] [40], the drugs used for this research were amoxicillin, paracetamol, ibuprofen, Bactrim, cetirizine, azithromycin, bacitracin, loratadine, xylometazoline [10], the data was collected from eight parenting web sites in a seven year period (2005-2012) [10], a total 1290 posts were collected, 900 annotated and 300 kept in reserve [10]. A database was created to keep drugs and posts, drugs were organized with their generic names and Anatomical Therapeutic Chemical (ATC) codes [10]. Used UMLS Metathesaurus [31] for biomedical vocabularies, and other resources such as European Agency for the Evaluation of Medical Products (EMEA), Medicines and Medical Devices Agency of Serbia (ALIMS) [41], the DrugBank Database [42], and MetaMap [10] [43], which is a lexical system that maps text to concepts in UMLS Metathesaurus. The following disproportionality measures were used to qualify drug safety: reporting ratio (RR), proportional reporting ratios (PRR) [10], reporting odds ratios (ROR) and information component (IC), each of these measures find the association between ADR and drug. The evaluation of this method resulted in precision = 75.3%, recall = 64.7%, and F-measure = 69.599% [10].

The research by [9] was unique as it tried to not only identify the ADR, but also identify when it was first mentioned [9], all using tensor decomposition to assist in the classification, having used tensors as data containers instead of matrices, greatly helping in missing data and helped identify some ADR mentions before the official FDA announcement [9]. The main advantage of this approach is that it does not require expert annotation for the data [9]. Matrix Based Technique (MBT) was compared to the proposed Tensor Based Technique (TBT).

External resources are used for item extraction, mainly Consumer Health Vocabulary (CHV) for drugs and ADR lexicon [9]. Temporal factor is considered important for ADR detection and monitor association rule [9], which is why the association will be made based on a time period of a year, with value p being time window, and q being overlapping year, several combinations of both were tired to compare with TBT [9]. After the association rule mining is done with temporal analysis for each ADR, a matrix of drug x time is made where each cell is the lift measure between ADR and Drug under specific year [9]. TBT introduces using a tensor (3D matrix) between drug, ADR and time, using CANDECOMP/PARAFAC (CP) decomposition technique [9]. Tensors allow the observation of dataset in 1 year time window without overlapping, so data from 2001 and 2002 are used in the same vector, averaging the results from both years [9]. Dataset about 20 drugs were extracted from 500 threads each drug from MedHelp [26] totaling 16344 threads ranging from 1997 and 2011 [9]. Evaluation measures were not determined, but apparently several ADRs were discovered by TBT as opposed to MBT, and some of the existing ADR were discovered to have been mentioned earlier than when it was alerted by the FDA [9].

2.2.6. Supervised Learning

Next is supervised learning approaches, most ambitious ADR detection system try to track them through twitter or a mix of twitter and other sources, trying to use the large data set that can be provided through twitter [11] [12] [13]. Among the three research SVM, Naïve Bayes, and Maximum entropy were the most commonly used and evaluated.

The first to create a model that follows this approach is [44], which tried to use an ensemble of classification algorithms to classify the dataset on several stages, namely Support Vector Machines (SVM) with RFB kernel and Naïve Bayes (NB), using messages in Yahoo! Forums as dataset (exact size not determined), it used two feature sets: general vocabulary, and meta-features with specialized lexicons from MedDRA [44] [45].

The research made in [11] tried to research the ability to use multi-source dataset corpus, mainly three, twitter with data provided from Arizona State University [24], using 10822 tweets. 1082 of which were annotated by experts using the Inter Annotator Agreement (IAA) using Cohen’s Kappa [11] [46]. DailyStrength [25] with 10617 posts were used (23.7 % contain ADR mentions). And the ADE corpus which contains 23516 phrases with 29% containing ADR mentions, the corpus is not social media but it was used as a proof of concept that the system

introduced can analyse data from any source [11]. Three supervised classifiers were made to test the data, Naïve Bayes (NB), Support Vector Machines (SVM), and Maximum Entropy (ME). To perform pre-processing, Porter stemmer was used using implementation by NLTK toolkit [11] [47], POS tagging was done using Stanford parser [37] and twitter parser [48]. UMLS concepts were identified using MetaMap [11] [43]. Synonymous terms were identified with WordNet [11] [38]. The features were by how often a change happens, so if an ADR is reduced, a good change happens and vice versa, so the feature set is built by identifying these changes using a sliding window on the phrase, the features are More-Good, More-Bad, Less-Good, Less-Bad [11]. Two other features are lexicon related derived from Leaman's [5] lexicon, the two features in question are the Boolean feature of presence or absence of ADR mention, and the numeric feature of the number of times an ADR is mentioned, they also collect a topic based feature using Mallet tool [11] [49], the features are the topic mentioned and the sum of the relevance score [11]. Other features include: length of the text, the presence of certain tags from Stanford parser. To implement Naive Bayes (NB) and Maximum Entropy (ME) Weka tools are used, for SVM they use LibSVM implementation [11]. The SVM uses RBF kernel. Several combinations of the dataset were used to test the program, giving varying results, but the after combining all the datasets and testing them on the test data, it outperformed all the others, the results are shown in this table [11].

Test Data	Training Data	ADR F-score	non-ADR F-score	Accuracy (%)	95% CI
ADE	ALL Three	0.799	0.913	87.8	86.8 – 88.7
TW	ALL Three	0.564	0.934	88.5	87.1 – 89.8
DS	ALL Three	0.686	0.887	83.4	81.7 – 84.9

Figure 2 Results from Literature survey N.1 [11]

Also not all features have shown improvement as indicated by this table [11]. Where removing a feature tends to cause a drop in performance, except Synonyms (Syn-set) and topic model, have limited to no effect on the datasets (Except ADE corpus) [11].

Features	TW		DS		ADE	
	Accuracy	ADR F-score	Accuracy	ADR F-score	Accuracy	ADR F-score
All	86.2	0.538	83.6	0.678	88.2	0.812
N-grams	80.7	0.424	82.6	0.654	85.9	0.775
UMLS STs and CUIs	85.7	0.505	82.8	0.652	81.9	0.711
Syn-set Expansions	86.1	0.545	84.0	0.669	87.9	0.778
Change Phrases	87.1	0.521	83.9	0.665	88.0	0.803
ADR Lexicon Match	86.1	0.492	83.5	0.663	86.1	0.780
Sentiword Score	86.2	0.530	82.8	0.659	88.3	0.805
Topics	86.1	0.535	83.7	0.670	87.6	0.801
Other Features	86.9	0.534	83.6	0.677	88.1	0.809

Figure 3 Results from Literature survey N.2 [11]

This research[12], Tried to not only identify ADRs but drug users, implementing two SVM classifiers, one to first identify users of five cancer drugs (still on trial) on Twitter , and the other to identify which ADR did they suffer [12]. The dataset acquired was is a 2 billion tweet collection collected using the Twitter API collected by [50] [51] for a different study [12]. For this study, the Tweets were reduced to four fields, 1) ID, 2) User ID, 3) Timestamp, and 4) text. The features extracted from the Tweets are textual and semantic [12], textual features include Bag-of-Words (BoWs), number of hasht-tags in the document, number of reply tags, number of negating words, the number of URLs, the number of pronouns, and the number of occurrences of the drug names or synonyms [12]. The semantic features are those derived from the UMLS [31] concepts extracted using MetaMap [43]. Trying to limit the data by choosing specific keywords more specifically the number of semantic type and group. After some trials, the Apache Lucene information retrieval library [52] parallelized with Amazon Cloud E2C was used. This lead to having the dataset limited to 239 potential users of the drug, 72 of them were confirmed. The evaluation of the classifiers was vaguely defined, but what is clear is that the first classifier had a prediction accuracy of 0.74 and an Area-Under the-Curve (AUC) of 0.82, and the second got an accuracy of 0.74 and AUC of 0.74 [12].

In [13], the works of the previous study was criticized over the technical aspect of using 72 sized dataset for both training and testing and not using the classifiers to classify more raw Tweets [13], it also criticises the lack of mentioning for the overall results, as well as using investigational drugs (still on trials) as it is not recognized whether the tweeter is using placebos

or the real drugs [13]. With that in mind, the drugs chosen (Duloxetine, Gabapentin, Baclofen, Glatiramer, Pregabalin) have been chosen for being in the market for a number of years. This time the Tweets were mined manually of the period of 80 days using Twitter API (Which does not allow searching for posts older than 2 weeks) for a total of 6829 tweets, removing matching brand names of the drugs to peoples' names and using only English tweets [13]. The features used were personal pronouns and sentiments derived from the NLTK [47], this is because the study required having "personal experience" tweets as opposed to "non-personal" tweets, according to the study; personal experience is expressed more often with pronouns. Using this principal, three classifiers were made, Naïve Bayes, SVM and Maximum Entropy (ME) [13]. ME was the most superior at precision= 0.866, Recall=0.842 and F-Measure=0.848, SVM's result was precision= 0.856, Recall=0.810 and F-Measure=0.820, and Naïve Bayes was precision= 0.858, Recall=0.827 and F-Measure=0.835. Using 600 tweets for training and 285 for testing [13].

2.2.7. Hybrid and Unique

Of mixed solutions, a mixed unsupervised and supervised learning [14] system (called ADRMine) had been developed. Data was provided by DailyStrength [25] and twitter using 81 drugs. The system used the same data provided in [24] which was used by [11]. Expert annotators annotated the posts and matched it with IAA's Cohen's Kappa [46], with the gold standard includes only reviews with complete IAA. The result is 4720 reviews from DailyStrength (+1559 test) and 1340 tweets (+444 test). An additional 313833 DS reviews and 397729 drug related tweets were gathered in a total 711562 postings to form an unlabelled set which would be used in unsupervised learning. A lexicon was generated using the previously mentioned tools UMLS [31] and SIDER [14] [32]. Concepts are extracted using Conditional Random Fields (CRF), a CRF classifier is used to extract ADR concepts from user sentences using CRFsuite implementation [53], turning them into individual tokens, beneficial effects were also identified as it was noted that including them improves performance of ADR extraction [14]. The CRF features extracted include context features: the 3 tokens before 3 tokens after and current token (Spelling correction was done with Apache Lucene [52]). ADR lexicon: a binary feature that shows whether or not current token is included in the lexicon [14]. POS: generated with Stanford parser. Negation: Features that indicated that the token is negated using syntactic

dependency rule [14]. The other feature extracted is the learning word embedding, the embedding is a meaningful real-valued vector of configurable dimension (between 50 and 500), these vectors were generated using Word2vec tool [54], which learns the embedding based on the word's contexts in different sentences, then a K-Mean clustering operation is performed to cluster the words into n (=150) different clusters (Each cluster has some common words, like one only including ADR or only including drug names or dates) [14]. Seven features are defined based on the generated clusters, which are the cluster numbers of the current token, the three preceding and the three following tokens. ADRMine's CRF classifier was compared to 4 extraction techniques (SVM, Lexicon-based, and two simple baselines based on MetaMap). The system proved superior to the other techniques, reaching an F-Score= 0.821 for a recall=0.784 and precision=0.860 for DS dataset and F-Score =0.721 for Recall=0.682 and Precision=0.765 for twitter. It was concluded that the lexicon, POS and negation features did not add a significant contribution with a huge dataset as it did with smaller dataset [14].

There was only one example of hidden Markov Model [15], the study describes three type of sources for ADR information, biomedical sources (books, journals, magazines, drug package labels) which may not be up to date, most accurate [15]. Clinical sources (patients' data) which are not free and are limited by ethical, legal and social constraints, may also be inaccurate. Online Forums are the most inaccurate, but they are the most numerous, up to date and totally free, refers to many other works found in the survey [15]. The data was extracted from Medication.com [28] and steadyHealth.com [55] using JSoup crawler, collecting 8065 posts from Medication and 11878 from SteadyHealth [15]. To extract relationships between entities, the information extraction module is made, consisting of Named Entity Recognition (NER) and Relationship Extraction (RE) sub-modules [15]. NER helps to identify entities of interest in a given text, such as names of drugs, side effects and keywords or phrases relating them together using lexicon based method [15]. The dictionary of drug names was crawled from the drug lists on drugs.com [56], side effects from SIDER [32]. RE than identifies the relationships between named entities using Hidden Markov Model, which learns the association between the drug name and side-effect in a given text [15]. The HMM is defined using the following parameters: N number of states, M number of observation symbols, A N by N transition probability matrix, B N by M observation probability matrix. Pi N by 1 initial state probability vector. Around 2000 annotated training data messages is used to train the classifier using the Baum-Welch training

algorithm [15]. The learnt model is used with Viterbi decoding algorithm to predict the hidden states for the observed sequence data in the testing set, where if the three states (drug, side-effect, connecting keyword) then the text is flagged as a positive drug/side-effect relationship. Using 10-fold-cross validation, the model got an F-score = 0.76, HMM however was unable to distinct between ADRs and symptoms of the drug, reducing the dictionaries themselves causes problems, especially for the drug dictionary [15].

Another unique approach was the Partially Supervised Learning technique (PSL) [16], which tries to solve the lack of large labelled data, by only giving the classifier a small number of labelled data and dynamically augmented it throughout the learning process, this would eliminate the need for a large expertly annotated dataset [16]. The dataset in question I extracted from Yahoo! Forums ProzacAwareness (Prozac drug) and Selective Serotonin Reuptake Inhibitors (SSRIsex), having a total of 6400 posts (1600 ADR and 4800 Non-ADR posts) [16]. A consensus detection is used to identify which new example goes to which cluster of data (positive or negative) using Rocchio algorithm [16]. An SVM and Naïve Bayes (NB) classifiers were created with and without PSL for comparison. SVM and NB scored less than 68 F-score without PSL, while the SVM and NB with PSL reach 89.74 and 86.32 F-score respectively [16]. The proposed approach was compared using benchmark labelling heuristics (EAT and PNLH) outperforming both of them in terms of F-Score [16].

Chapter 3: Methodology and Implementation

The methodology of this project is separated into several parts.

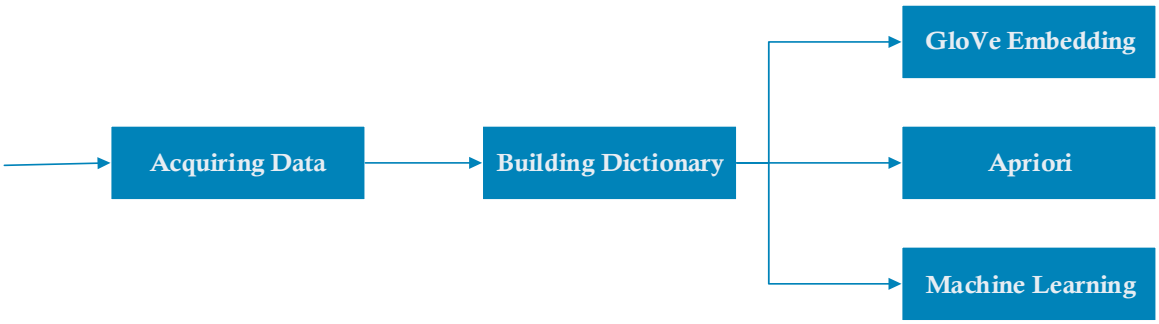


Figure 4 Data flow from acquiring to usage

3.1. Acquiring the dataset

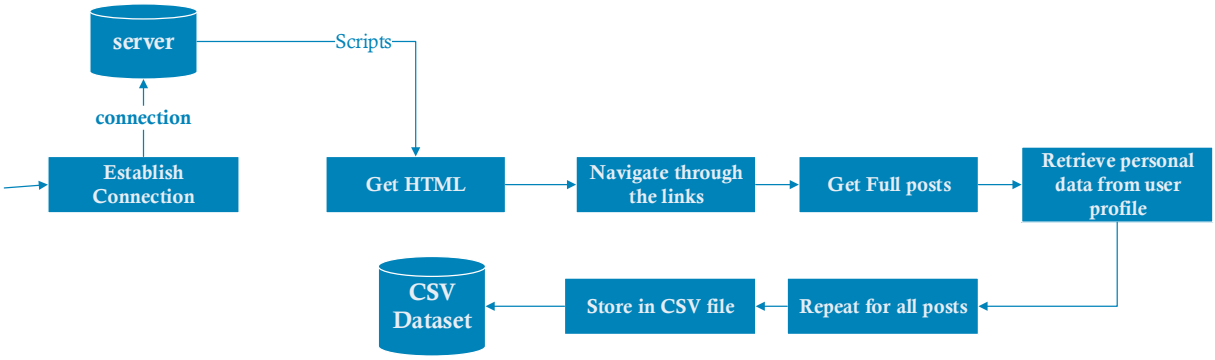


Figure 5 Acquiring dataset using crawler

Acquiring the dataset was made possible by creating a data crawler using Java JSoup library using the following steps:

1. Get the establish connection to server.
2. Retrieve HTML scripts.
3. Use HTML hyperlink tags to navigate and retrieve post pages.
4. Find pages with the tags that contain the full text.
5. Get user profile pages which includes some user profile data (age and gender).
6. Find the personal data tag, split it into parts, take the age part and the gender part (which is marked by integers and the words male and female)
7. Repeat and store the data in an array list of data objects.

8. Store the dataset into CSV files that can be easily accessed later (switched to excel later because it is even more convenient).
9. Handle connection problems during the crawling process by Backing up the data.

3.1.1. Subjects to consider

- Choosing the Drugs for this research: For the purpose of this research, drugs using for chronic diseases had to be chosen. After some consideration, drugs used for hypertension were chosen, and they are Lisinopril, Nadolol, Amlodipine, Diltiazem, Hydrochlorothiazide and Atenolol.
- Choosing the medical forums for mining: The websites chosen for this purpose are MedHelp and AskAPatient, which was chosen since most of its posting members are more committed to share their personal data, such as age and gender. The dataset acquired from both forums will be used comparing results and quality of the datasets.
- Finding the correct links and tags: JSoup establishes a connection to the server and returns the HTML script as text to a variable, and from that script tags can be chosen based on IDs or classes, given a universal search query link (example: https://www.medhelp.org/search/expanded?cat=posts&page=2&query=Nadolol), the web can easily be navigated through JSoup, and given the correct tags from each the given posts (example: subject_msg), data can extracted from each page and it's HTML script.
- Store Data: The chosen data storage is on CSV file which can be accessed using MS excel, they can also be used later using Pandas library in Python, and later saved as excel files (which proved even more convenient than CSV files).
- AksAPatient: Using crawler was neither necessary nor possible, the site was protected against crawling activities. However its data was in a table format which was much easier to simply copy and paste, a script was later made to remove all the problems in the text format. AskAPatient is more consistent comparing to MedHelp, however user data is impossible to extract other than age and gender, however all data extracted is correct and therefore could be used for comparison against MedHelp dataset.

3.1.2. Data retrieval

Several natural language processing techniques were implemented using Python to extract the data necessary, using NLTK (Natural Language Tool Kit) library

- Tokenize data: Turn the words into separate tokens.
- Remove stop-words: Stop-words like (and, a, or) were removed to decrease the size of data.
- Stemming data: Porter stemmer was used to turn words into their roots (exhaustion, exhaustive, exhausted= exhaust), both the original and the stemmed tokens were kept into separate csv columns.

3.2. Build Dictionary

A dictionary filled with concepts like ADR, Disease and Mental issues were needed to narrow down the search premise into the UMLS, using the following steps:

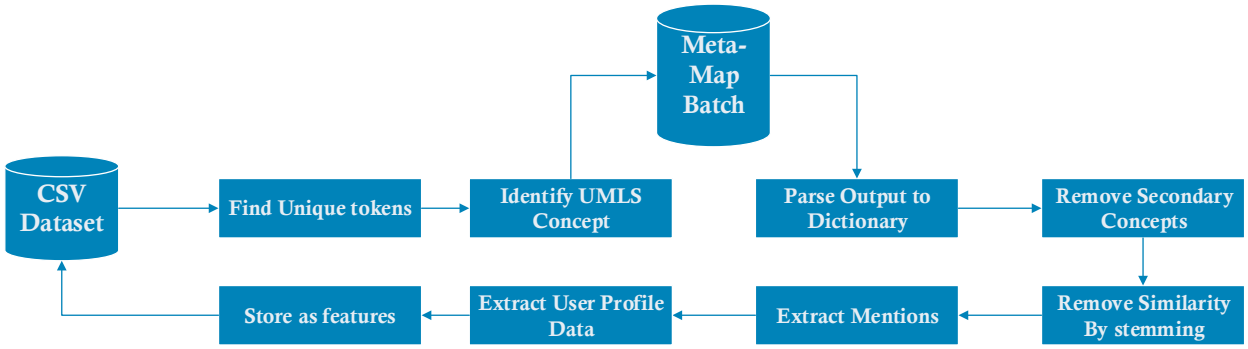


Figure 6 Steps for building dictionary

1. Find and store all unique tokens.
2. Identify UMLS concepts using MetaMap.
3. Parse MetaMap output into a python dictionary format.
4. Remove secondary concepts.
5. Stem the token to remove similarity.
6. Extract mentions from the Dataset using the dictionary.
7. Extract missing user profile data as features.
8. Store mentions as features and classes for classification.

3.2.1. Subjects to consider

- Find term frequency: This in itself is not necessary for building a dictionary, but the TfidfVectorizer from Sklearn library can double as a retrieving method for all unique words in the text, term frequency will be used later, but for now all unique words are stored in two term frequency files, one for stemmed words and one without stemming.
- MetaMap batch: MetaMap as mentioned before is used to extract UMLS concepts, by sending a file to the batch system, which was necessary due to size of the dataset.
- Stemmed or Un-stemmed: The stemmed version was not effective in extraction because it removed the meaning of the words (“Acne” became “acn” which means nothing and was therefore undiscovered). Therefore the un-stemmed version was instead used for discovery, with the terms being stemmed later, which reduced the size of the dictionary by insuring that similar words are not repeated (confusion, confused = confus).
- Extract the concepts from MetaMap: MetaMap output needed is classified into three categories ([Signs and Symptoms] = ADR, [Disease or Syndrome], and [Mental or Behavioral Dysfunction]), a script was made to handle MetaMap output and extract these concepts into files containing the concept in Python Dictionary Format.
- Extract The concepts per post: The dictionary was copied and pasted into a python script (After some minimal manual revision) to be used for concept extraction from the dataset, every post had was scanned for any token that match any concept in the dictionary, and these concepts were than aligned it’s related meaning ('cramp': ['Muscle Cramp ', 'Cramping sensation quality ']) or meanings, this arrangement is made so that concepts with similar meanings don’t get repeated (Pain = Pain, Ache = Pain).
- Secondary concepts (where the same word could mean several things and they appear in other meanings as well) were removed if they were similar and don’t add meaning, if repeated, the meanings were unified to decrease useless data.
- Age was also extracted by applying a moving window of three tokens that searches for some limited age related words couple with numbers (I am 28 years old). As well as blood pressure (150/70).

3.3. Unsupervised Algorithms

3.3.1. Association with GloVe

Word Embedding: A technique for language modeling and feature learning [57], which transforms words in a document into a continuous real number [57], traditional way to find calculate it include simple term frequency, TFIDF and co-occurrence matrix [58]. Newer methods however include the use of neural networks, such methods include Word2Vec [58].

GloVe: Global Vectors for Word Representation [59], is a very popular library created by Stanford to be used for word embedding based on co-occurrence matrix. It is used here to make some indications which can be observed later in the machine learning phase. Word embedding shows the association between all the words in the corpus.

GloVe works as follows [60]:

- Find the co-occurrence matrix probability of each word x [59] [60].
- Calculate the context of the words together based on the distance x+= 1/distance [59] [60].
- Take the log for each value in the matrix (+1 to avoid 0 values) [59] [60].
- Weight value x in the matrix from 0 to 1 based on a function ($f(x) =(x /xmax)^a$) [59] [60].
- Calculate the final Co-occurrence matrix using the cost function [59] [60]:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w}_k + b_i + \tilde{b}_k - \log X_{ik})^2,$$

Figure 7 cost function for the GloVe Co-occurrence matrix

- The matrix can be calculated over many dimensions, creating many vector values that can be calculate to find the distance between two certain words using cosine or Euclidean distance [59] [60].

Implementation Details:

- Both stemmed and un-stemmed datasets were used to create a word vector models, the stemmed dataset turned out to be more effective as it generalized some terms instead of repeating them (confused, confusion=confus).
- The models were saved as 2D array for vectors in .txt format.
- The text file was loaded into tensorflow projector [61] [62]to represent the points of the model into an intelligible 3D (in truth 100D) plot (pictured above).
- This plot allows the observation of related concepts in a readable manner, each dot representing one of the model's label. The closer two dots are to each other, the higher their association, and therefore the probability of co-occurrence.

Two approaches were made for the analysis:

1. Discover the relation between ADRs and Drugs.
    - After the models were saved into several formats, they were trimmed to only include ADRs and Mental issues, as they are the most relevant for this analysis.
    - By looking at the closest vectors to a certain drug, it was possible to find which ADRs and Mental issues have the highest chance to occur when using the drug. For example, amlodipine was found more related to hoarseness, frenzy and hallucinations than it is related to pain, nervousness and alcohol abuse. It is therefore more expected for a patient to encounter hoarseness- for example- than to encounter pain while taking the drug.
2. Discover any related concepts to the drugs and ADRs.
    - The models were trimmed to include the words with a relatively high term frequency (above 20, 40, 60, or 100), in order to make it more readable by removing the least used words with little effect on the model.
    - The goal is to find any randomly related concepts that might relate to the use of the drug, like for example dosage, age, a certain height or weight. It could therefore be understood from the results if there is any remote relation between the usage of this drug, or the presence of the ADR, and the presence of these other concepts.
    - The dataset was slightly modified to attach several concepts to each other to become on term (200 mg = 200mg) which limits the model size and helps make it more accurate and readable.

3.3.2. Association with Apriori

Association rule mining is extremely important as a data mining technique, it can be help for decision making. And in the context of this project, it can identify the relation between the occurrences of Drugs, Drug Family and all the ADRs, mental issues and diseases.

In association rules, the goal is to identify the following major components [63]:

- Support: Records Containing A/Total Records = the number of times one or more items appear in the records [63].
- Confidence: Confidence (A =>B) = (Records containing both (A and B)) / (Records containing A) = the likelihood that if one item appear in a record, another one also appears [63].

Apriori Algorithm is an application to the association rule mining used to decrease the run time for a large dataset [63] [64]. Generally, the standard algorithm is calculate all the rules for all the dataset. In Apriori however:

- Set a minimum requirement for any component, support as a beginning [63] [64].
- Calculate the support for all 1-itemsets (I1) [63] [64].
- Choose I1 that meet the minimum requirement as candidates (C1), and drop the rest [63] [64].
- Find I2 for C1 and calculate the rest of the components [63] [64].
- Repeat for Ix and C(x-1) until no more candidates are available [63] [64].
- Typically, the biggest length is C3 at I4 sets, in practice a limit to the length can be placed.

Implementation Details:

Apyori [63] library was used for this implementation on python.

- As input, concepts mentioned were put into a list of lists, each list corresponding to a record of the posts (example: ["Hypothyroidism ", "Hypertensive disease ", "Ulcer "]).
- Each list was extended to include age, gender, drug and drug family, in order to obtain the relation between them (Drug and Drug Family relations were later removed to avoid confusion.
- After running Apriori using Apyori. The result was printed in a .txt file to be reviewed later.

3.4. Supervised Learning

3.4.1. Preparing the dataset

After building the dictionary, it is now possible to identify the diseases, ADRs, and mental issues that were mentioned in the user posts.

- **Pandas** library was used to access and manipulate the data.
- A scanner was made to iterate on every token in every record in the stemmed version of the dataset, matching each token with an equivalent in the dictionary.
- When a token matches the dictionary, it marks the meaning of the concept as existing if another token with same meaning, such as (‘ache’: ‘pain’, ‘pain’: ‘pain’), it is ignored to limit repetition.
- The number of concepts (diseases, ADRs and mental issues), were counted for each record.
- Two types of classes are created using this process.

    1. Concept exists, where for each concept, a Boolean value is given to determine the existence of the concept in the record.
    2. Concept count, where the number of specific concepts in a particular range is drawn, meaning the number of distinct ADRs in a record for example is 5, this number is recorded and then assorted in the following ranges as a class.
        - 0 for 0 concept.

- 1 for range [1,3].
- 2 for anything more than 3.

In the previous example, 5 will be in the 2 class.

1. As for the features, user information harvested is used, such as age, gender, blood pressure, weight and height. Unfortunately, the MedHelp dataset has a lot of missing data, so not all user information could be used at the same time.
2. To solve this issue, the dataset was divided into three groups.
    1. Age + Gender only
    2. Age + Gender + Weight + Height
    3. Age + Gender + Blood Pressure

Dataset Description	MedHelp Complete	Weight/Height	Blood Pressure	Ask A Patient
Number of Posts	1557 Posts	130 Posts	462 Posts	757 Posts
Maximum Post size	12951 Token	7868 Token	6829 Token	1939 Token
Minimum Post Size	8 Token	254 Token	174 Token	3 Token
Features	Age, Gender, Drug, Drug Family	Age, Gender, Drug, Drug Family, Weight and Height	Age, Gender, Drug, Drug Family, Blood Pressure	Age, Gender, Drug, Drug Family

Table 1 Dataset Description

Feature Description	MedHelp Complete	Weight/Height	Blood Pressure	Ask A Patient
Age	12 to 107 Blank=515	15 to 79 Blank=10	12 to 80 Blank=107	15 to 91 Blank=0
Gender	'Male': 581, 'Female': 553, Blank: 423,	'Male': 71, 'Female': 41, Blank: 18	'Male': 199, 'Female': 167, Blank: 96	'Female': 423, 'Male': 333
Weight		27 to 172 KG		
Blood Pressure			Max: 295/135 Min: 102/60	

Table 2 Feature Description

- Now that missing data is in an acceptable level, the remaining missing values (Age and Gender) were imputed using the SciKit learn library, SimpleImputer. Gender was imputed based on most frequent strategy, while age was imputed based on median strategy.
- The data sizes for the three datasets respectively is: 1557, 130, 462. With dataset 2 $\subseteq$ dataset 1 and dataset 3 $\subseteq$ dataset 1. Each subset is stored in a separate excel file.

- The classes labels were stored in separate excel files, from which they can be extracted later and used in the classifier.
- The same procedure was applied for AskAPatient dataset, however the dataset did not include anything other than age and gender, therefore as a whole the size of the dataset is 757 with no divided parts between them.
- The dataset can thus be modified to suite any learning model, either by including all the labels (except the class) as a part of the features, or by separating them completely and only using the patient profile and drug/drug family as features.
- Drug/Drug family can also be used as features, where the goal is to find out the possibility by which a drug could cause ADRs and diseases.

3.4.2. Dataset Imbalance

All the dataset have data label imbalance due to infrequency in the ADRs per each patient, in the complete MedHelp dataset, the highest percentage is “Pain” label, with 76% negative and 24% positive. The subsets are generally more even, but there is still too much imbalance in most labels. The “Hypertensive Disease” label and count labels are however have a better distribution, so they could be used for a more accurate measure of the quality of the dataset.

As for the remaining labels, several techniques could be applied.

1. Oversampling re-use: Several records of the minority class are repeated [65].
2. Under-sampling: Several records from the majority class are unused [65].
3. SMOTE (Synthetic Minority Over-sampling Technique): Where entirely new instances of the records are created to fill the gaps [65].

Approach number 3 was chosen for this application, using the imbalanced-learn implementation [66], using SMOTENC library (NC = Nominal Continuous).

SMOTE works as follows [65]:

1) Isolate the minority class [67]

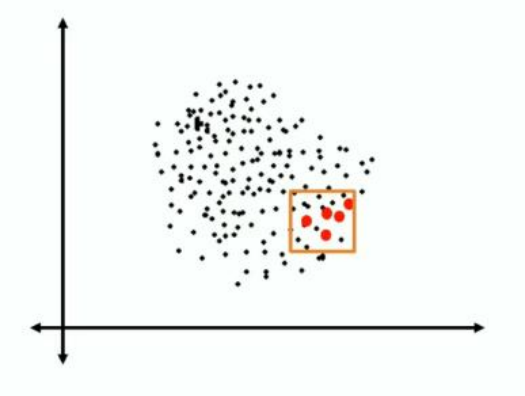


Figure 8 SMOTE: All Data [67]

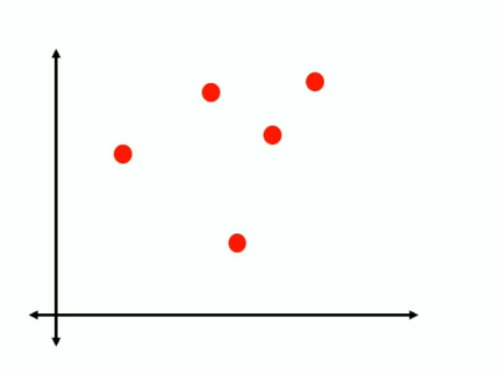


Figure 9 SMOTE isolated minority [67]

2) Find the k nearest neighbors depending on the over sampling requirement, calculate the distance between the two neighbors [67].

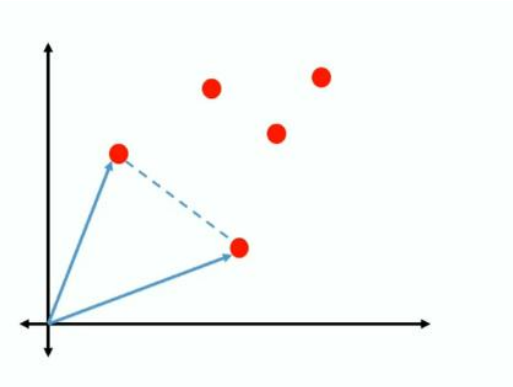
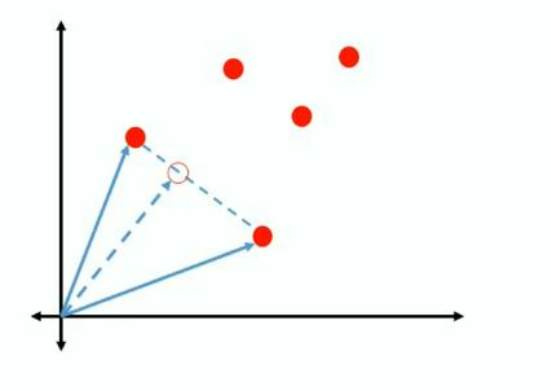


Figure 10 SMOTE KNN [67]

3) Multiply the distance with a random number between 0 and 1, placing a new record of the minority class on the new point created [67].

Figure 11 SMOTE random new record by distance [67]



4) Repeat until data label imbalance is solved [67].

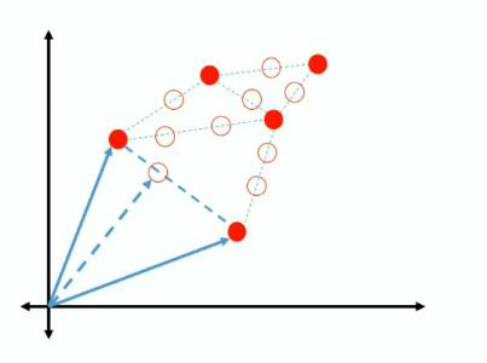


Figure 12 SMOTE new Data [67]

SMOTE however only works with continuous features, which is problematic since most the dataset is nominal/categorical, using either Boolean features or discrete categories that describe drug/drug family and counts.

This is why **SMOTENC** is used instead [65], SMOTENC looks acts the same way as SMOTE, however, between the closest neighbors, the median of the standard deviation of the nominal feature is taken into consideration for the distance [65]. With the distance calculated, continuous values are calculated based on normal SMOTE, while nominal features are given based on the majority of values within those neighbors [65].

SMOTENC implementation works exactly the same as SMOTE, by fitting the dataset to each other using SMOTENC object, only the difference is pointing out which of these features are nominal, SMOTENC is compatible with Pandas Dataframe, with the a list of new features as output to be used in the classifier.

Performance of the classifiers with and without SMOTENC will be compared together.

3.4.3. Preparing Classifiers

3.4.3.1. Goals:

There are several goals that could be achieved by using classifiers.

1. Predict the possibility for a user to be affected by an ADR or Disease based on extracted profile.
2. Predict which drug or drug family is the cause of the ADRs or Disease affecting the user, with the help of the user profile.
3. Predict the number of ADRs or Diseases that could be caused by a drug, based on the user profile and/or existing cases.

A separate classifier group will be made for each goal, each classifier group using a different dataset arrangement, which will be furthermore elaborated on in the following section.

3.4.3.2. Dataset Arrangements:

Features differ depending on the used subset in terms of user profile.

Arrangements	Labels	Features
1	Labels, Counts are categorized while running. One Label is chosen for each prediction.	User Profile, drug and drug family, profile differs for
2	Labels are only limited to identifying which drug or drugfamily was responsible for any given disease.	Features include labels from the arrangement 1 and the user profile.
3	Labels and Features are mixed, with the wanted feature being removed from the comparison dynamically.	

Table 3 Data arrangements used for the experiments

Arrangement 1 and 3 will include predictions for the Count of ADRs diseases, mental issues count will be ignored as their mentions are very low for in askapatient dataset. However the mentions themselves will be used for as features.

3.4.3.3. The Classifiers

Three classifiers are made as a performance comparison.

3.4.3.3.1. Random Forests

3.4.3.3.2. SVM

3.4.3.3.3. Naïve Bayes

3.4.3.3.1. Random Forests (RF)

Random Forests is an ensemble classifier for classification and regression trees [68] [69].

It works as follows:

Create a certain number of decision trees classier N [69] [70].

Commence bootstrapping operation:

- Subsets of the original dataset are randomly selected to be used for each tree classifier in N, repetitions are possible.
- For each subset, choose random features to use instead of using them all [69] [70].

Each resulting tree can be completely different from the others and can give different classification results.

Each one of these results calculated in a vote, with the highest voted result being chosen as the final result for the classification [69] [70].

4.2.2.2. Random Forests in Scikit Learn

The random forests implementation is provided is SciKit learn as an ensemble classifier, the main parameter given is “n_estimators”, which is the number of bootstrapped trees that should be constructed. Estimators between 100 and 1000 were tested to ensure accuracy.

4.2.2.3. Feature Selection

In addition to the classification capabilities, RFs can be used for embedded feature selection [71], as each time a new tree is made with a subset, a purity metric can be measured, since not all trees see all features, there is an assured de-correlation between all features, this also makes them less prone to overfitting [71]. By calculating the purity of the tree can be used to derive the importance of the features, giving a standard on which feature is more important than the other and thus which feature could be removed [71]. Scikit Learn provides this capability view feature_selection library, which can use an RF classifier to test the data and choose the best features based on an initial classification [71]. This method will be tested on all the classification methods, and observations will be provided for the improved performance if any.

4.2.2.4. SVM

First introduced by Vapnik, SVM is a very popular classification and regression technique [72]. It is based on the Structural Risk Minimization principle (SRM) [72], where the classifier maps input vectors to a higher dimensional space where a maximal separating hyperplane is constructed on each side of the hyperplane separating the data [72]. The goal is to maximize the hyperplane, the assumption being that when the bigger distance between two hyperplanes are maximized, the better the generalization error of the classifier [72].

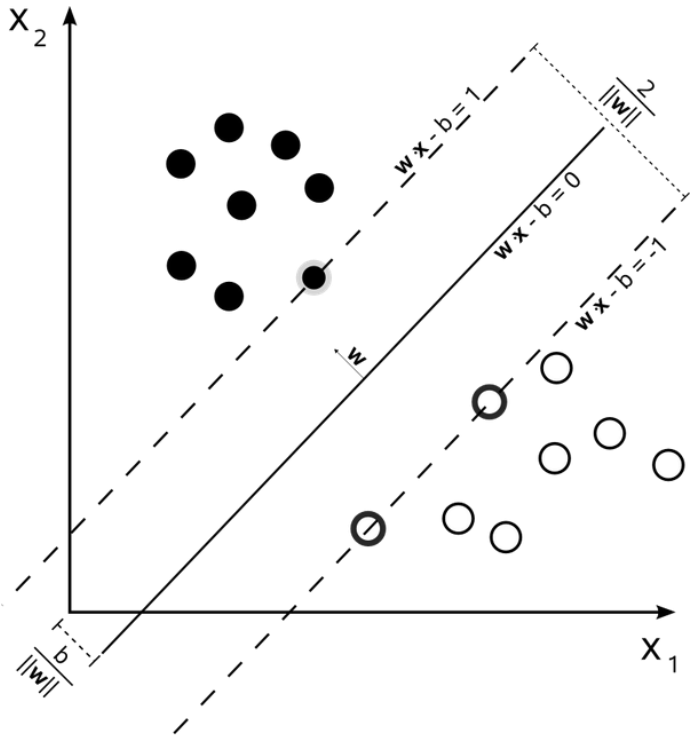


Figure 13 SVM [77]

The goal is to maximize 2 / | W |

The hyperplane is calculated based on several factors, the most important of which is kernel. Here are the most popular kernels [72]. RBF

- Linear
- Sigmoid
- Polynomial

RBF being the most commonly used for handling higher dimensional space better than Linear, with less parameter than Polynomial, and less numerical difficulties [72].

SciKit learn implementation provides all these assortment of kernels, as a parameter to the SVM classifier [72], with RBF as default kernel.

4.2.2.5. Naïve Bayes Classifier

Naïve Bayes (NB) is one of the simplest probabilistic classifiers available, it is based on Bayes rule [73]. Where P© is the probability of a class (yes or no/Pain exist or doesn't exist), and P(X) is the probability of all the features. P (C|X) is the probability of the class given all the features used [73]. If the probability of P(C=exist|X) > P(C=not|X), than the classifier predicts that yes, this ADR exists and vice versa [73].

$$P(C|\mathbf{X}) = \frac{P(C)P(\mathbf{X}|C)}{P(\mathbf{X})} \qquad P(C|\mathbf{X}) = \frac{P(C)\prod_{i=1}^{n} P(X_i|C)}{P(\mathbf{X})}.$$

Figure 14 Bayes Rule [73]

NB is reported to be extremely simple as it only relies only on simple mathematics and statistics.

The implementation on SciKit learn is GaussianNB, which uses the Gaussian distribution to represent continuous variables. Although there are other distribution options are available [74]. Including Bernoulli, Multinomial and Complement.

$$\frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Figure 15 Gaussian distribution [74]

Performance Metrics: The performance of the classification is important to test capabilities of the trained classifier and evaluate the results without having to delve into specific cases [75]. The performance metrics used are based around the concept of the confusion matrix [75].

		Actual	
		Positive	Negative
Predictions	Positive	TP	FP
	Negative	FN	TN

Figure 16 Confusion matrix

Each cell identifies the count of test cases that received a given result. True positives and negatives (TP and TN) indicate that the prediction was correct and matching to the actual test data. While false negative and positive (FN and FP) indicate the wrong in predictions. Usually it is better to have FP than FN because a positive can be an alert, while negative means no detection at all [75].

Based on the matrix, four metrics were chosen to measure the performance:

1. Accuracy = (TP+TN)/(FP+FN+TN+TP): Correct prediction/all predictions [75], it is good measure when most classes are nearly balanced [75].
2. Precision = TP/(TP+FP): The amount of TP in proportion to all the positives [75].
3. Recall = TP/(TP+FN): The amount of Negatives that were missed [75].
4. F1-Score= 2*Precision*Recall/(Precision + Recall): the harmonic mean of precision and recall, meant to be used as a simplification, instead of having to look into 2 different values, F1-Score gives an approximation that is closer to the smaller number, and thus more accurate and representable number [75].

There are other metrics that could be used, but these are the most standard (as discussed in the literature survey).

4.2.2.6. Testing Parameters:

To calculate the performance, the dataset has to be trained using a given set of data, and tested against another set it has never seen. The dataset can be used for that by randomly splitting the data into a testing set and training set, the usual ratio for this purpose is 70% for training and 30% for testing. Values in the confusion matrix can be calculated in the testing part, and from this the metrics can be calculated.

4.2.2.7. Implementation details

The implementation was originally meant to be made in python 2.7, and indeed most the text processing made Data retrieval and Dictionary Building was in 2.7. However the initial experiments with GloVe showed that 2.7 was extremely outdated. Therefore the learning process was moved to python 3.6, in which installing was much easier.

Environment: Using python required setting up an environment that could allow the use of all necessary libraries, thus Anaconda was installed, and configured for both 2.7 and 3.7 while each was in use.

This small list of libraries were necessary for the making of this project.

- Pandas: File and Dataset management
- NLTK (PorterStemmer, remove stopwords, tokenize text)
- Sklearn (classifiers, Count Vectorizer, label encoder, feature selection, K-Fold testing, imputer, TFidf Vectorizer)
- Imblearn.SMOTENC
- GloVe

4.2.2.8. IDE

Most text processing was made on anaconda's Spyder IDE, however, while making the classifiers, the implementation was moved to Jupyter Notebook, which was more convenient as it allowed complete code separation and easier document comments.

Jupyter uses separated cells which can be modified and run separately, while still maintaining the same variable and importation pool. This allowed the interfacing between several functions, where each classifier is in a different cell, each run of the classifiers is in a different cell, the dataset manipulation and preparation is independent from running the classifiers.

Therefore whenever any arrangement was tested, the only change needed was to the data manipulation functions with minimal changes to the classifier.

Jupyter does not support concurrency. However for convenience, all cells could run at once in the sequence they were placed.

Cells could be changed from code to headings or simple text, they could be added, moved, stopped and repeated independently from one another.

4.2.2.9. Classifier Flow

As shown in this flow chart:

1. The dataset is collected from the excel files.
2. The dataset is modified to suit whichever dataset arrangement in use.
3. Then there are two steps:
    - SMOTENC resampling.
    - Feature Filtering with random forests.
4. Store metrics to be used for comparison between all combinations later on.

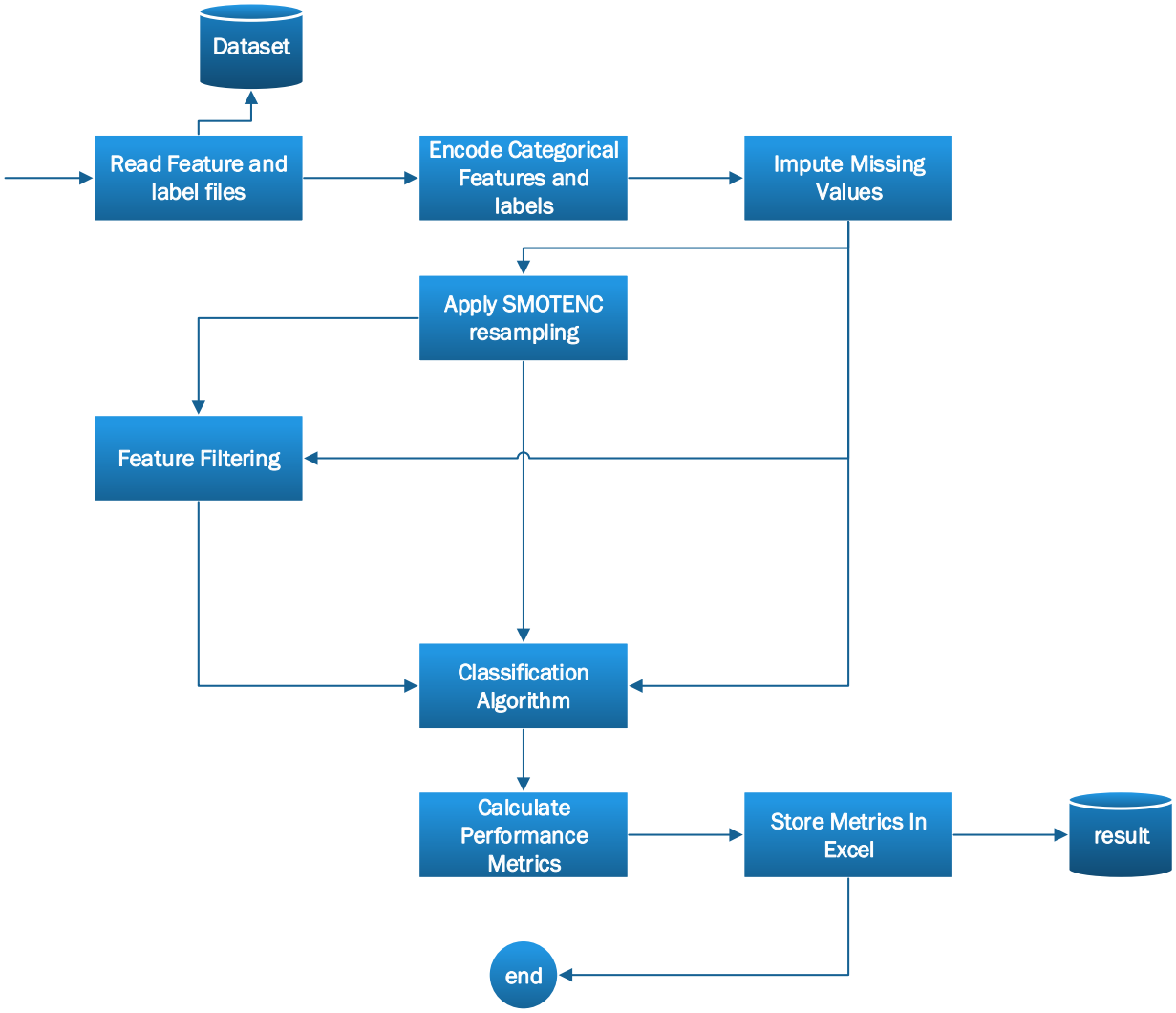


Figure 17 Classifier flow

Chapter 4: Results and Discussion

4.1. Unsupervised Algorithm

4.1.1. GloVe Results

In this section, the results given from applying GloVe to the dataset will be discussed, where the closest terms given will be provided for context. All the experiments were run on a 100 dimensional parameter.

4.1.2.1. Drugs and ADR relation

Using the vectors acquired from GloVe, the following results were obtained:
The tensorflow projector [62] allowed displaying the results in a 3D view. By searching for a single word, mainly a drug or an ADR, the words closest to the meaning of the word are highlighted. The tool allows decrease the viewed portions to an intelligible level, this will be limited to 40 words. The distance between the words can be calculated with either Euclidean or cosine distance, which don't make much difference, therefore Euclidean distance was used, it should be noticed that the words are stemmed.

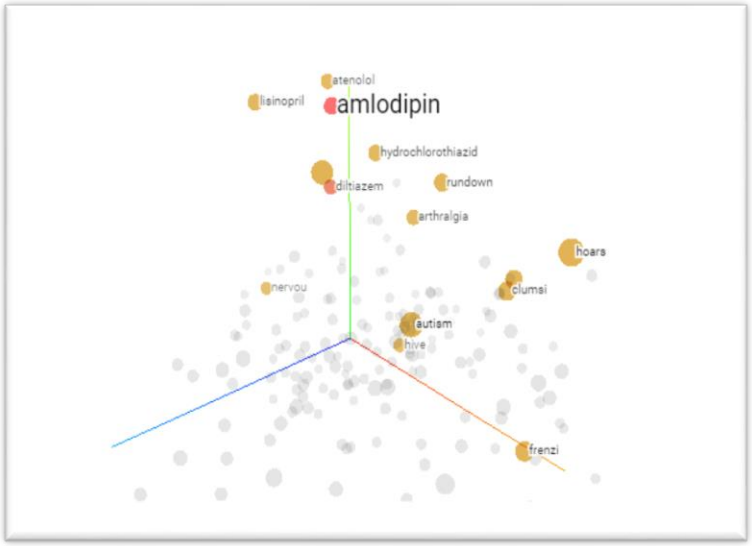


Figure 18 Tensorflow vector projector

4.1.1.2.1. MedHelp Dataset:

GloVe was able to identify which concept are the drugs, where for all the drugs, the closest concepts are in majority the five other drugs. It also shows the most frequent ADRs that were encountered in the dataset.

Despite Amlodipine and Diltiazem being of the same drug family, they have little ADRs in common, which could suggest that the problem is in the drugs themselves, not in the drug family. Atenolol and Nadolol on the other hand had more in common although in different degrees to the other.

Amlodipine	Diltiazem	Hydrochlorothiazide
diltiazem 0.889 atenolol 1.104 lisinopril 1.117 hydrochlorothiazid 1.126 hoars 1.173 rundown 1.231 clumsi 1.257 apnoea 1.271 autism 1.271 arthralgia 1.280 stubborn 1.281 nervou 1.288 frenzi 1.289 hive 1.296 tremor 1.296 dereal 1.299 doe 1.301 icd 1.311 analgesia 1.313 sluggish 1.313	amlodipin 0.889 lisinopril 1.164 atenolol 1.259 flush 1.262 fluctuat 1.270 qol 1.282 confus 1.285 tremor 1.291 cramp 1.293 nervous 1.300 hydrochlorothiazid 1.307 breathless 1.307 sick 1.318 insomnia 1.319 nervou 1.320 frenzi 1.320 stiff 1.322 ailment 1.324 aphasia 1.324 ssd 1.327	lisinopril 1.122 amlodipin 1.129 atenolol 1.147 nadolol 1.180 rundown 1.203 hoars 1.239 stubborn 1.251 autism 1.261 arthralgia 1.276 sigh 1.278 rigor 1.288 anxieti 1.289 hemiplegia 1.292 hangov 1.294 heartburn 1.294 clumsi 1.296 convuls 1.296 psychot 1.297 cramp 1.301 lethargi 1.304

Table 4 GloVe Result (a)

The Common ADRs that repeat most commonly are:

- Hoarseness
- Apnoea
- Sleeplessness (Insomnia)
- Anxiety
- Doe

Nadolol	Atenolol	Lisinopril
atenolol 1.128 hydrochlorothiazid 1.180 seizur 1.205 lisinopril 1.244 tens 1.277 psycholog 1.289 constip 1.291 doe 1.297 suffoc 1.306 despond 1.307 miser 1.309 nightmar 1.310 psych 1.312 anxieti 1.323 quiver 1.324 sleepless 1.331 spray 1.340 apnoea 1.340 flashback 1.341 hyper 1.344	lisinopril 1.078 amlodipin 1.113 nadolol 1.128 hydrochlorothiazid 1.147 apnoea 1.191 arthralgia 1.202 rundown 1.207 anxieti 1.214 hoars 1.241 sleepless 1.250 suffer 1.257 autism 1.271 diltiazem 1.271 headach 1.280 ssd 1.311 anxious 1.315 doe 1.317 depress 1.322 tens 1.323 rigor 1.328	atenolol 1.078 hydrochlorothiazid 1.122 amlodipin 1.127 diltiazem 1.175 apnoea 1.193 nadolol 1.244 cough 1.245 autism 1.253 rundown 1.273 block 1.274 depress 1.280 dizzi 1.283 ssd 1.288 anxieti 1.288 confus 1.294 hoars 1.296 exert 1.307 withdraw 1.311 convuls 1.318 sleepless 1.320

Table 5 GloVe Result (b)

4.1.1.2.2. Ask A Patient Dataset:

This dataset was more focused than MedHelp, and therefore its size was much smaller, especially that only two drugs were tested on it.

Despite this, it not only identified, several common ADRs, but it also identified other drugs that are not within the scope of the dataset search.

Atenolol		Lisinopril	
suffer	1.186	cough	1.233
hydrochlorothiazid	1.214	atenolol	1.245
apnea	1.237	suffoc	1.246
jerk	1.244	headach	1.259
tingl	1.245	sick	1.261
lisinopril	1.245	hydrochlorothiazid	1.271
spot	1.249	dizzi	1.272
nightmar	1.283	doe	1.273
snore	1.302	tire	1.277
sore	1.305	amlodipin	1.277
dizzi	1.307	anxieti	1.282
miser	1.316	swell	1.286
rash	1.318	hallucin	1.292
fatigu	1.321	ach	1.294
suffoc	1.322	jerk	1.299
anxieti	1.329	spot	1.302
vertigo	1.337	bloat	1.310
flush	1.337	hoars	1.311
stiff	1.340	ill	1.314
hangov	1.340	fatigu	1.315

Table 6 GloVe Result (c)

4.1.2.2. Related Concepts Discovery

4.1.1.2.1. MedHelp Dataset:

The previous section discussed the results of the limiting presented vectors to drugs and ADR. In this section, the vectors presented are limited by their overall frequency, the goal is to discover what kind of general terms accompany drugs, to insure that drugs are mentioned, they are attached to posts before being trained as well as the drug family.

The results show that it was still capable of identifying other drugs, but less effectively, drug family is always the closest, since it is attached to wherever a drug is mentioned, the most common things people appear to mention in their posts is the dosage (x mg), the number of times they take a dose (twice), the duration (daily), and the injection method (pill or tablets), by far the most common word is 'Hi', which shows how this medical forum is aimed more towards community rather than a focused drug review website.

Amlodipine		Diltiazem		Hydrochlorothiazide	
calcium	0.893	calcium	0.923	diuret	0.759
diltiazem	0.937	amlodipin	0.937	25mg	0.999
5mg	0.973	channel	1.155	50mg	1.067
10mg	1.067	metoprolol	1.204	10mg	1.124
30	1.134	prescrib	1.208	5mg	1.175
50mg	1.162	syndrom	1.214	prescrib	1.188
Hi	1.166	120	1.223	20mg	1.188
25mg	1.178	specif	1.224	twice	1.194
child	1.192	D	1.227	12	1.194
daili	1.193	chanc	1.228	amlodipin	1.200
these	1.198	blocker	1.239	with	1.203
current	1.199	daili	1.239	daili	1.211
hydrochlorothiazid	1.200	XL	1.242	lisinopril	1.212
chanc	1.204	Hi	1.242	Hi	1.217
norvasc	1.206	10mg	1.246	hctz	1.220
20mg	1.209	ace	1.247	tablet	1.226
atenolol	1.209	lisinopril	1.250	100mg	1.227
twice	1.211	tachycardia	1.253	nadolol	1.231
100mg	1.218	current	1.255	pill	1.234
lisinopril	1.220	sudden	1.257	dear	1.234

Table 7 GloVe Result (d)

Nadolol	Atenolol	Lisinopril
beta 0.938	beta 0.885	angiotensin 0.798
20mg 1.123	50mg 0.972	10mg 1.043
clonidin 1.133	25mg 1.004	20mg 1.045
Hi 1.175	Hi 1.129	5mg 1.075
later 1.180	100mg 1.152	hctz 1.105
atenolol 1.208	put 1.164	40mg 1.113
blocker 1.210	40mg 1.176	Hi 1.150
40mg 1.216	daili 1.183	put 1.155
made 1.217	lisinopril 1.186	25mg 1.158
50mg 1.218	10mg 1.198	prescrib 1.177
began 1.226	take 1.201	atenolol 1.186
hydrochlorothiazid 1.231	old 1.201	daili 1.202
twice 1.233	blocker 1.205	take 1.205
ace 1.237	dose 1.205	hydrochlorothiazid 1.212
infect 1.239	hypertens 1.208	hypertens 1.215
includ 1.243	nadolol 1.208	dose 1.219
40 1.250	amlodipin 1.209	amlodipin 1.220
rather 1.251	treat 1.216	took 1.229
physician 1.252	5mg 1.220	diuret 1.230
diuret 1.257	dear 1.222	recent 1.232

Table 8 GloVe Result (e)

4.1.1.2.2. Ask A Patient Dataset:

The previous analysis from the dataset is further confirmed, where dosage is even more frequent, however it is more discrete and to the point, with most words written being directly related to drug consumption and related issues to it and indicate more suffering compared to the community friendly MedHelp.

This does not necessarily mean that MedHelp users face any less issues, but it shows that AskAPatient is efficient and compact, with less descriptions given comparing to MedHelp.

This however makes AskAPatient harder for data mining as it contains less user information that users mention while talking about their issues.

Atenolol		Lisinopril	
beta	0.883	angiotensin	1.003
25mg	1.137	blood	1.177
100mg	1.171	high	1.189
50mg	1.185	hctz	1.193
still	1.202	thi	1.200
It	1.214	take	1.203
definit	1.217	specif	1.206
heart	1.220	pressur	1.209
work	1.221	48	1.221
suffer	1.224	took	1.221
bit	1.224	10mg	1.225
4	1.230	side	1.227
lower	1.231	start	1.229
flutter	1.242	stop	1.236
left	1.245	effect	1.240
later	1.247	switch	1.245
ER	1.248	prescrib	1.245
coupl	1.250	My	1.246
take	1.250	lower	1.248
palp	1.251	never	1.249

Table 9 GloVe Result (f)

4.1.2. Apriori Results

In this section, the results obtained from applying Apriori algorithm to the dataset's mentioned ADRs, diseases, mental issues, gender of user and drug used, will be discussed. The parameters specified were made as follows:

- Minimum support = 6%, this value was chosen based on the 6% of patients going to hospitals for because of ADRs.
- Minimum confidence = 65%

4.1.2.1. MedHelp

Rule: ['Amlodipine'] -> ['Hypertensive disease ']
Support: 0.06767676767676768
Confidence: 0.6836734693877552
==========
Rule: ['Pvc'] -> ['Atenolol']
Support: 0.10808080808080808
Confidence: 0.656441717791411
==========
Rule: ['Pvc'] -> ['Beta']
Support: 0.11515151515151516
Confidence: 0.6993865030674846
==========
Rule: ['Common Cold '] -> ['Upper Respiratory Infections ']
Support: 0.06262626262626263
Confidence: 0.96875
==========
Rule: ['Diabetes'] -> ['Hypertensive disease ']
Support: 0.0696969696969697
Confidence: 0.6571428571428571
==========
Rule: ['MUNGAN SYNDROME '] -> ['Hypertensive disease ']
Support: 0.15858585858585858
Confidence: 0.6796536796536796

==========

Rule: ['MUNGAN SYNDROME ', 'Female'] -> ['Hypertensive disease ']

Support: 0.06161616161616162

Confidence: 0.6630434782608695

4.1.2.2. Discussion on MedHelp

It can be noted that the only drugs mentioned within the given parameters are Atenolol and Amlodipine, being a beta-blocker and a calcium channel blocker respectively [76].

Not much information were obtained from MedHelp using Apriori within those constraints, except as a proof of concept, by finding related diseases (['Common Cold '] -> ['Upper Respiratory Infections ']) and disease cured by the mentioned drugs (['Pvc'] -> ['Atenolol'] and ['Amlodipine'] -> ['Hypertensive disease '])

4.1.2.3. AskAPatient

Rule: ['MICROCEPHALY , EPILEPSY , AND DIABETES SYNDROME ', 'Coughing'] -> ['Female']

Support: 0.06551059730250482

Confidence: 0.7555555555555555

==========

Rule: ['MICROCEPHALY , EPILEPSY , AND DIABETES SYNDROME ', 'Lisinopril'] -> ['Female']

Support: 0.10211946050096339

Confidence: 0.7162162162162162

==========

Rule: ['Angiotensin', 'MICROCEPHALY , EPILEPSY , AND DIABETES SYNDROME ', 'Coughing'] -> ['Female']

Support: 0.06358381502890173

Confidence: 0.75

==========

Rule: ['MICROCEPHALY , EPILEPSY , AND DIABETES SYNDROME ', 'Lisinopril', 'Coughing'] -> ['Female']

Support: 0.06358381502890173

Confidence: 0.75

==========

4.1.2.4. Discussion on AskAPatient

Lisinopril is the only drug that appeared in this result with those parameter, AskAPatient was more complete and concise comparing to MedHelp, therefore better detailed patterns about the ADRs, Diseases and drugs appeared.

The most interesting thing however is that all the results include a females as a consequence of the other terms appearing (example: ['Angiotensin', 'MICROCEPHALY , EPILEPSY , AND DIABETES SYNDROME ', 'Coughing'] -> ['Female']), which is strange because only 55% of AskAPatient users are female, which could show that females in general are more in danger of getting a disease, and even more in danger of being inflicted by an ADR after consuming a drug.

4.1.2.5. Discussion on Apriori

The results from association rules are usually more respected when used on higher constraints, the given constraints in this case are either too large to find a useful statistical relationship with an acceptable statistical performance.

4.2. Supervised Learning Results:

In this section, various datasets and the arrangements used in them to compare the performance between the prediction capability of the classifiers Random Forests, SVM and Naïve Bayes, the primary point of comparison will be FScore and accuracy, with recall and precision being available to interpret the results from FScore.

The main point of comparison is the performance when SMOTENC and RF filter is applied, and the performance when they are not.

4.2.1. Legends

Dataset	Meaning
Complete Dataset	The complete original 1577 posts taken from MedHelp, including all the missing data and only having age and gender as user data
Weight/Height Dataset	Subset of Complete made of 130 posts which includes Weights/Heights extracted from the posts, some Heights are imputed.
Blood Pressure Dataset	Subset of Complete made of 461 posts the include blood pressure readingss
AskAPatient	The dataset obtained from AskAPatient, which includes 757 drug reviews.

Table 10 Legends to describe the Datasets

4.2.1. Summarized Discussion Supervised Learning

4.2.1.1. SMOTENC: Before and After

A comparison between the datasets before and after applying SMOTENC was made in the first experiment. The results provided show that not applying SMOTENC leads to higher accuracy, but had a smaller values for the other more important metrics, namely Recall, Precision and F1-Score (FScore). However for classes where Datasets are already balanced (like Hypertensive disease in Weights/Heights dataset and in the complete dataset) the results may not change much and may even drop when in use, this is why it is not recommended to use SMOTE unless the unbalance is overwhelming.

	SMOTE		No Smote	
	Accuracy	Fscore	Accuracy	Fscore
Complete Dataset	0.592814	0.59254	0.598290598	0.59287
Weights/Heights Dataset	0.729167	0.726196	0.820512821	0.78419
Blood Perssure Dataset	0.756098	0.756061	0.748201439	0.699413
AskAPatient Dataset	0.720548	0.71901	0.77092511	0.514319

Table 11 SMOTE VS NO SMOTE Hypertensize Disease

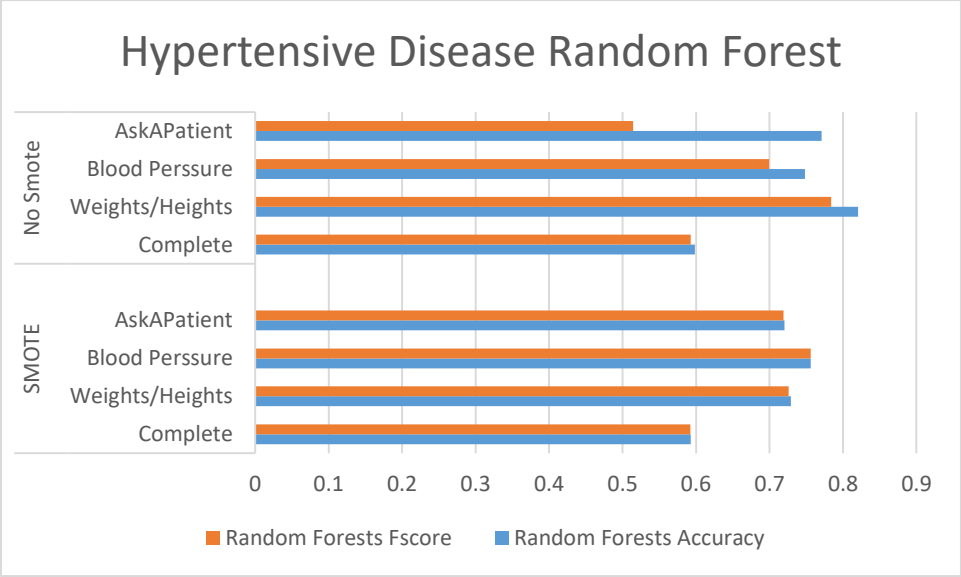


Figure 19 SMOTE VS NO SMOTE Hypertensize Disease

4.2.1.2. Random Forests vs SVM vs Naïve Bayes

Throughout the first and second experiments, Random Forests outperformed both SVM and Naïve Bayes, both when using SMOTENC and without, and even when using feature filtering. Therefore it was decided to ignore it on the third experiment if it did not produce a better result, which it did not.

Table 12 RF vs SVM vs Naive Bayes

	Random Forests		SVM		Naïve Bayes	
	Accuracy	**FScore**	**Accuracy**	**FScore**	**Accuracy**	**FScore**
Complete Dataset	0.53405	0.532069	0.443548	0.432164	0.305556	0.273662
Weights/Heights Dataset	0.629032	0.617931	0.387097	0.390853	0.451613	0.414229
Blood Pressure Dataset	0.620072	0.616403	0.512545	0.510708	0.365591	0.333318
AskAPatient	0.802139	0.802116	0.754011	0.752994	0.721925	0.714067

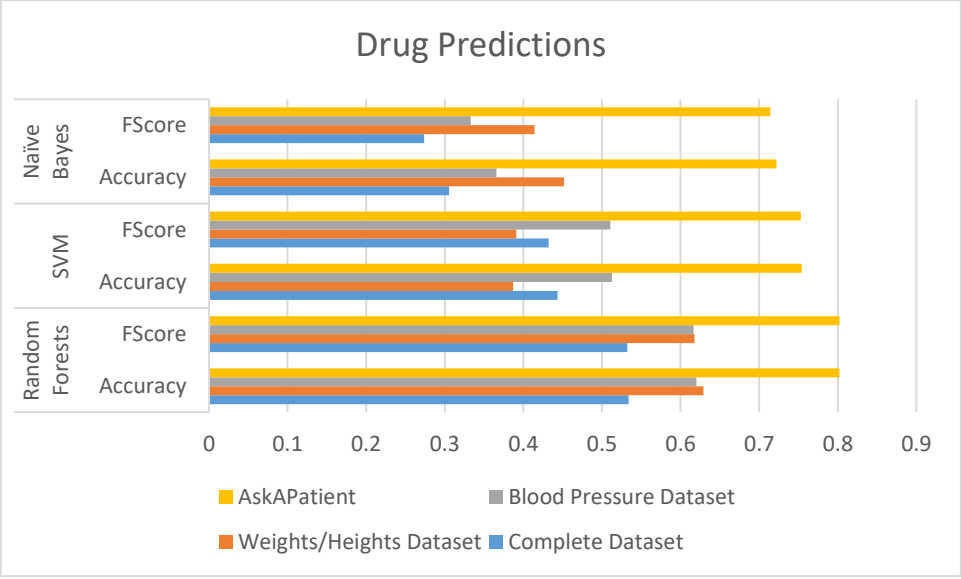


Figure 20 RF vs SVM vs Naive Bayes

4.2.1.3. Feature Filtering

Feature filtering using random forests in general produced varying results, both positive and negative in all metrics, but generally there was definitive improvement over not using it. However it did reveal a strange result, where after reviewing the second experiment, it turns out that the count of diseases, and mental issues were a major factor in improving the results in some cases, and that it was chosen as the most important besides user age.

Table 13 Filetring vs No Filtering

	With Filtering		Without Filtering	
	Accuracy	**FScore**	**Accuracy**	**FScore**
Complete Dataset	0.335125	0.333261	0.53405	0.532069
Weighted/Height Dataset	0.66129	0.648689	0.629032	0.617931
Blood Pressure Dataset	0.594982	0.585986	0.620072	0.616403
AskAPatient	0.828877	0.827277	0.802139	0.802116

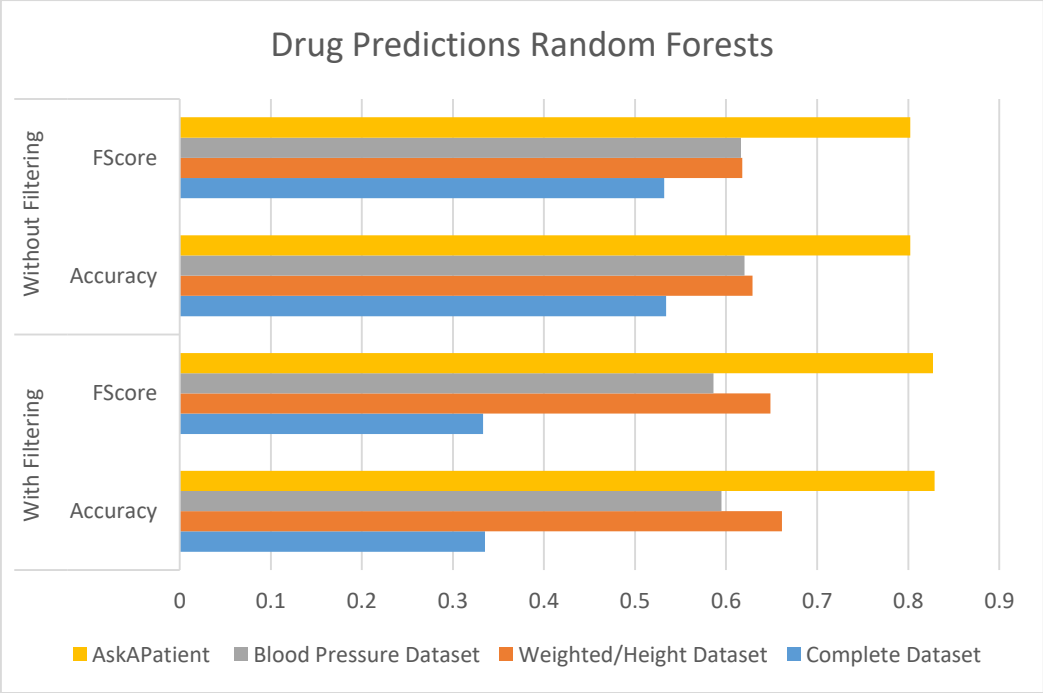


Figure 21 Filtering vs No Filtering

4.2.1.4. Predicting the potential number of ADRs and Diseases per user

Feature filtering did not choose ADR count as important, and after trying to find it in the first experiment the results were not satisfactory. On the other hand, trying to find the amount of diseases was more fruitful, for blood pressure and weights/heights datasets, which show how important they are in determining the potential healthiness of a person, but in any case, the results from the other classifiers make it discouraging to make such prediction.

Number Of ADRs	**Accuracy**	**FScore**
Complete Dataset	0.448882	0.449865
Weighted/Height Dataset	0.454545	0.455265
Blood Pressure Dataset	0.594203	0.594168
AskAPatient	0.574324	0.577359
Number Of diseases	**Accuracy**	**FScore**
Complete Dataset	0.494253	0.482369
Weighted/Height Dataset	0.655172	0.651343
Blood Pressure Dataset	0.713675	0.703807
AskAPatient	0.564033	0.572111

Table 14 Number of Diseases and ADRs

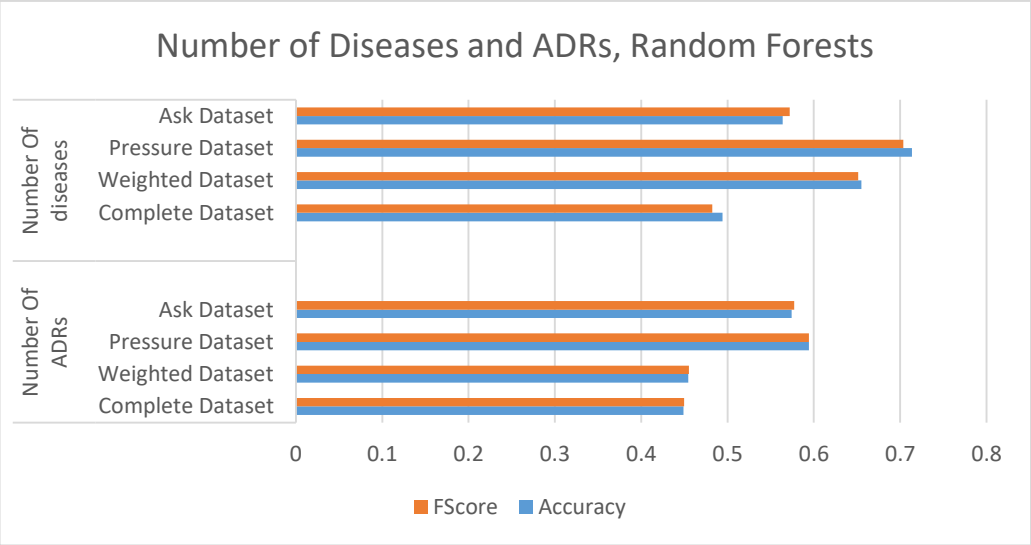


Figure 22 Number of Diseases and ADRs

4.2.1.5. Discovering ADRs and Diseases

The first and third experiments tried to predict the presence of the ADRs and diseases based on user profile, at 66% F-Score for Pain in MedHelp, the first experiment was successful with good results, however when using medical history from other ADRs and disease mentions in the third, this result has improved by more than 10%.

With Medical History	**Accuracy**	**FScore**
Complete Dataset	0.828816	0.828562
Weighted/Height Dataset	0.925926	0.922078
Blood Pressure Dataset	0.880597	0.880357
AskAPatient	0.79661	0.79661
Without Medical History	**Accuracy**	**FScore**
Complete Dataset	0.666191	0.664272
Weighted/Height Dataset	0.796296	0.794535
Blood Pressure Dataset	0.746269	0.745361
AskAPatient	0.714689	0.708642

Table 15 Pain Prediction with and without medical history

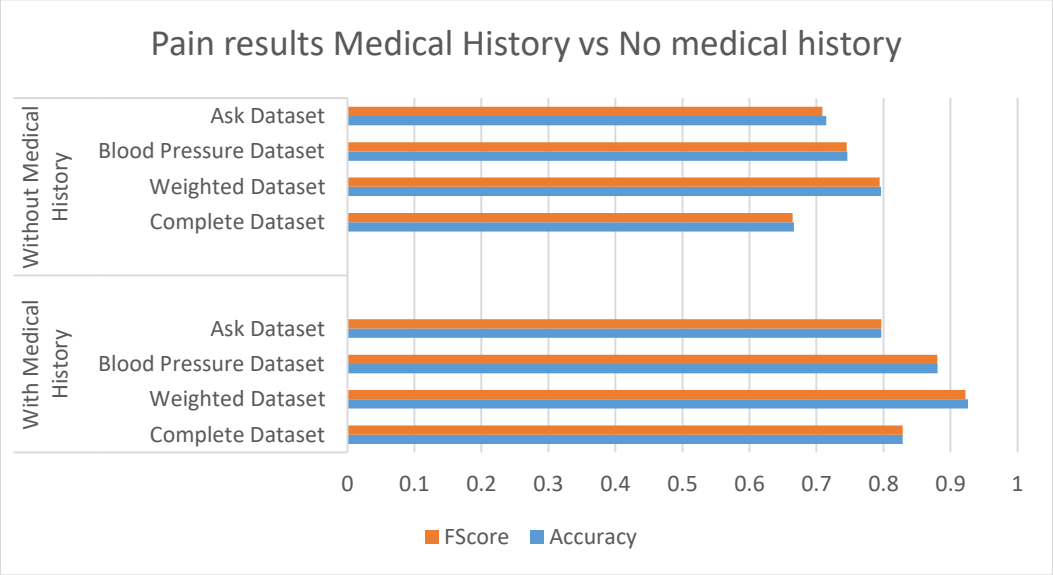


Figure 23 Pain predictions with and without medical history

4.2.1.6. Predicting the presence of any possible ADR

As a comparison with some related work, this prediction was made, the goal is to find if it is possible to predict if the user got afflicted by any ADR, no specifics given other than the presence, this has proven to be successful and comparable with some results given in some of the papers on supervised learning, more specifically the result in figure 2 and figure 3, even the results without applying SMOTE were satisfactory, and they show the potential of user profile and medical history as features.

No Smote	**Accuracy**	**FScore**
Complete Dataset	0.788462	0.772765
Weighted/Height Dataset	0.666667	0.547725
Blood Pressure Dataset	0.863309	0.82936
AskAPatient	0.823789	0.778104
Smote	**Accuracy**	**FScore**
Complete Dataset	0.843072	0.842752
Weighted/Height Dataset	0.892857	0.892308
Blood Pressure Dataset	0.85641	0.856406
AskAPatient	0.89645	0.89588

Table 16 ADR prediction

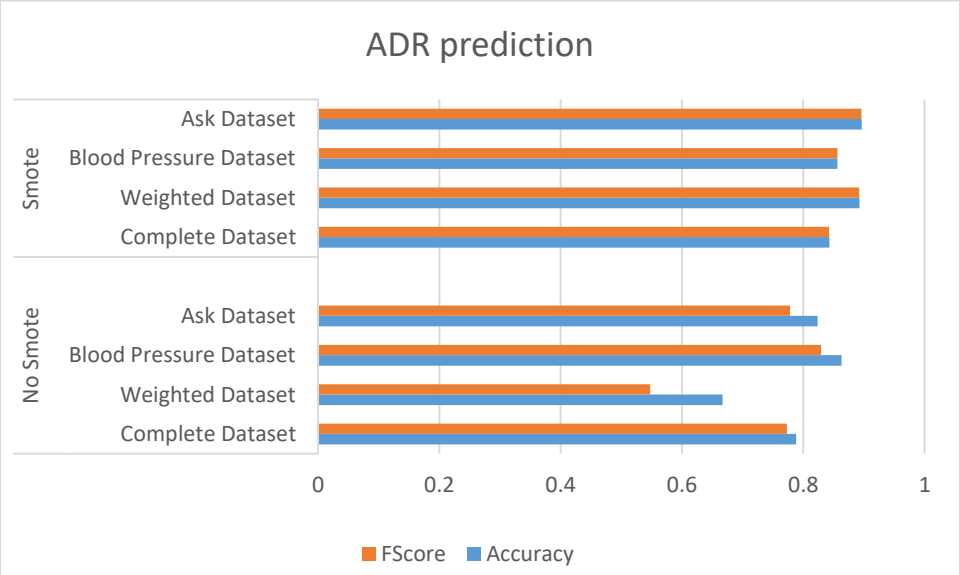


Figure 24 ADR prediction

4.2.2. First Experiment

- The most common ADR (Pain) as labels.
- The most common disease (Hypertensive Disease) as labels.
- Counts for ADRs, diseases and mental issues as labels.
- Testing the best number of estimators in terms of F-Score for random forests was made by trying all estimators between 100 and 1000 at 100 increment.

Table 17 Feature Description

Labels/Datasets	Hypertensive Disease	Pain	ADRCount	DiseaseCount
Complete Dataset	0: 834, 1: 723	0: 1168, 1: 389	1: 695, 0: 560, 2: 302	1: 870, 2: 489, 0: 198
Weight/Height Dataset	1: 80, 0: 50	0: 89, 1: 41	2: 48, 1: 45, 0: 37	2: 64, 1: 58, 0: 8
Blood Pressure Dataset	1: 273, 0: 188	0: 334, 1: 127	1: 229, 0: 137, 2: 95	1: 259, 2: 155, 0: 47
AskAPatient	0: 608, 1: 148	0: 590, 1: 166	1: 493, 0: 194, 2: 69	1: 407, 0: 324, 2: 25

4.2.2.1. Random Forests

SMOTENC versus without SMOTENC

Hypertensive diseases Predictons:

- Hyper Tension successfully predicted with blood pressure, with weights/heights being second. This gives importance to both readings as they can predict certain diseases.
- On trials without SMOTENC, the roles were reversed, weights/heights performance increasing much higher than blood pressure, this indicates that the unmodified pressure subset is inefficient compared to weights.
- While Ask a patient's performance fell, MedHelp complete dataset had remained stable, which is a direct result from the imbalance in ask a patient compared to MedHelp. The results however are a good indication that Hypertension can be predicted with a decent accuracy based on age alone, with other factors such as weight and height aiding the predictions.

Datasets	Accuracy	Precision	Recall	FScore
Complete Dataset	0.592814	0.593139	0.592869	0.59254
Weights/Heights Dataset	0.729167	0.739564	0.729167	0.726196
Blood Pressure Dataset	0.756098	0.756061	0.756061	0.756061
AskAPatient	0.720548	0.718645	0.720076	0.71901

Table 18 RF Hypertensive disease with SMOTENC

Datasets	Accuracy	Precision	Recall	FScore
Complete Dataset	0.598291	0.599863	0.595724	0.59287
Weights/Heights Dataset	0.820513	0.777778	0.792208	0.78419
Blood Pressure Dataset	0.748201	0.727473	0.68956	0.699413
AskAPatient	0.770925	0.520792	0.514936	0.514319

Table 19 RF Hypertensive disease without SMOTENC

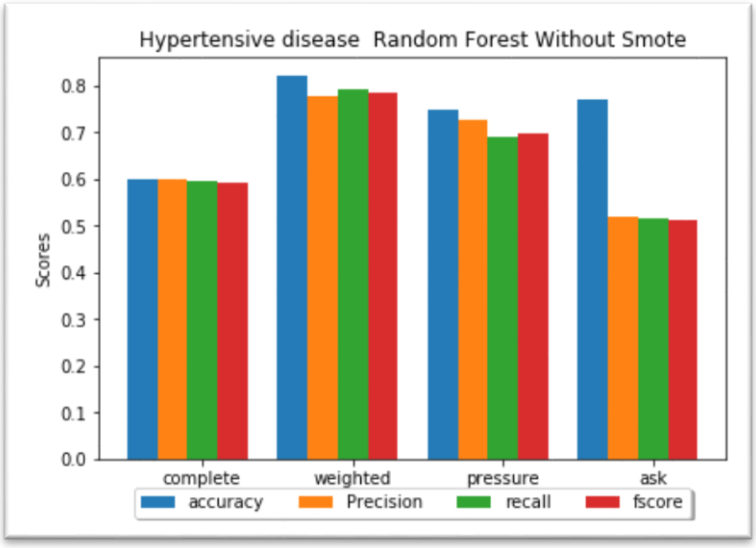


Figure 26 Hypertensive Disease RF Without SMOTE

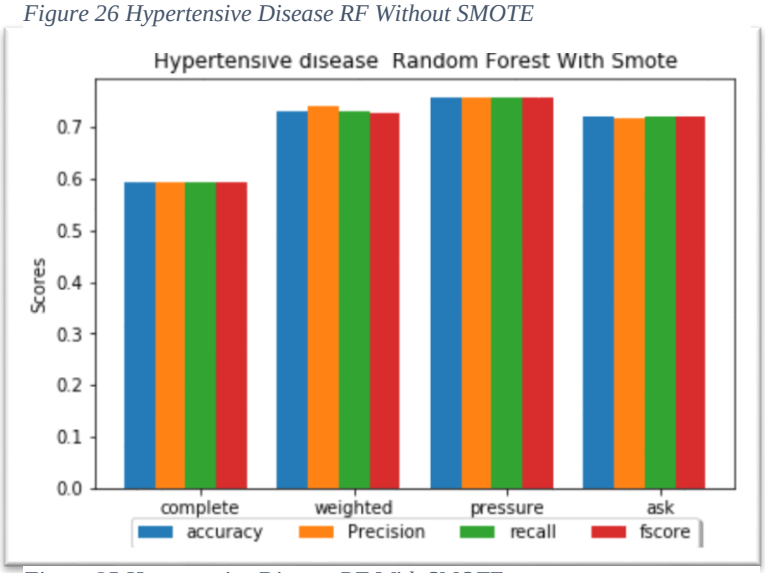


Figure 25 Hypertensive Disease RF With SMOTE

Pain Predictions:

- For pain, weights/heights have the biggest effect, which could indicate that pain as an ADR is cause by weights gain in relation to height.
- On trials without SMOTENC however, all metrics drops far below the accuracy, which reflects the terrible imbalance in all the datasets, despite the fact that it is the most mentioned ADR at 25 % positives to 75% negatives in MedHelp Complete dataset.
- Ask a patient has suffered the most, with a drop from 0.7 Fscore to 0.47.

Datasets	Accuracy	Precision	Recall	FScore
Complete Dataset	0.666191	0.665535	0.664104	0.664272
Weights/Heights Dataset	0.796296	0.807586	0.827941	0.794535
Blood Pressure Dataset	0.746269	0.745526	0.745233	0.745361
AskAPatient	0.714689	0.718844	0.708472	0.708642

Table 20 Pain with SMOTENC

Datasets	Accuracy	Precision	Recall	FScore
Complete Dataset	0.709402	0.552477	0.536667	0.536188
Weights/Heights Dataset	0.717949	0.621429	0.555195	0.546032
Blood Pressure Dataset	0.733813	0.690711	0.652751	0.6622
AskAPatient	0.696035	0.473708	0.483679	0.471291

Table 21 RF Pain without SMOTENC

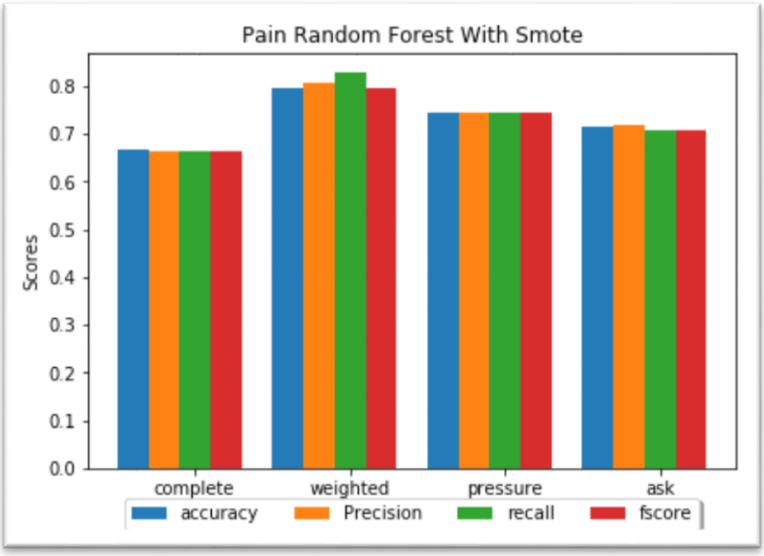


Figure 27 RF Pain with SMOTENC

Number of ADRs/ Diseases per user predictions:

- It can be observed that the number of ADRs that a person might have is much harder to predict compared to the number of disease.
- Weights/Heights are terrible for identifying ADR Count, however it performs slightly better for disease count without SMOTENC. Blood pressure decreased the least for ADR count, which could indicate the role it plays in causing ADRs in general.
- It is unclear whether the reason for the terrible performance is due to data imbalance solved by SMOTENC, or that it is naturally impossible to discover ADR and disease Counts based only on these factors.

Datasets	Accuracy	Precision	Recall	FScore
Complete Dataset	0.448882	0.455145	0.448323	0.449865
Weights/Heights Dataset	0.454545	0.50864	0.452381	0.455265
Blood Pressure Dataset	0.594203	0.597802	0.592951	0.594168
AskAPatient	0.574324	0.587815	0.574266	0.577359

Table 22 RF ADR Count with SMOTENC

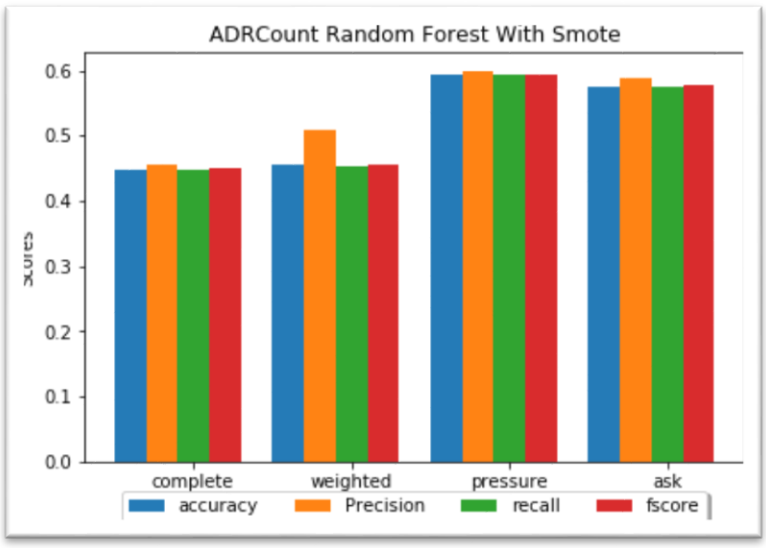


Figure 28 RF ADR Count with SMOTENC

Datasets	Accuracy	Precision	Recall	FScore
Complete Dataset	0.494253	0.489404	0.488716	0.482369
Weights/Heights Dataset	0.655172	0.657018	0.649832	0.651343
Blood Pressure Dataset	0.713675	0.715905	0.71548	0.703807
AskAPatient	0.564033	0.587314	0.565223	0.572111

Table 23 RF Disease Count with SMOTENC

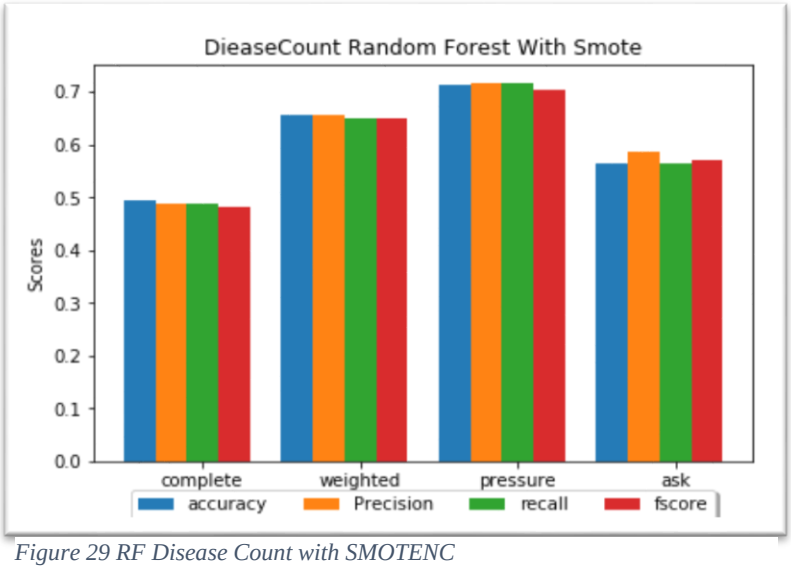


Figure 29 RF Disease Count with SMOTENC

Feature Filtering versus Without Feature Filtering

In this section, the comparison between the previous results before and after applying random forests filtering was applied, the conclusion is that it has very little effect on the results for this experiment, this can be attributed to the use of the very limited features. This point will be further tested with experiments 2 and 3, when all the remaining labels will be used as features. The following are some example that show how filtering performed with and without SMOTENC.

Hypertensive Disease Prediction

Ask a patient dataset and blood pressure subset have slightly improved their performance. The complete dataset and weight/height subset have decreased their performance slightly, but not far off from the original none filtered features.

Dataset	Accuracy	Precision	Recall	FScore
Complete Dataset	0.58483	0.590532	0.585076	0.578583
Weights/Heights Dataset	0.708333	0.708333	0.708333	0.708333
Blood Pressure Dataset	0.79878	0.802252	0.79942	0.798413
AskAPatient	0.734247	0.740344	0.7225	0.724099

Table 24 Hypertensive Disease RF with Feature Filtering and with SMOTENC

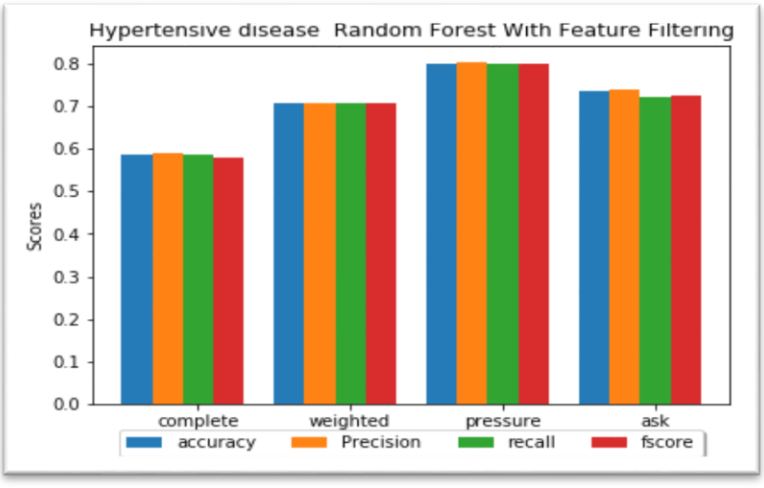


Figure 30 Hypertensive disease RF with Feature Filtering and SMOTENC

4.2.2.2. SVM

Pain Prediction

- SVM again showed much less performance compared to random forests, and again showing a drop without SMOTENC.
- It should be noted however that while the FScore decreased, the Accuracy is actually higher than RF. Which could mean that SVM could predict unbalanced data more accurately compared to RF, even if the overall performance is lacking.

Dataset	Accuracy	Precision	Recall	FScore
Complete Dataset	0.64194	0.643159	0.643231	0.641937
Weights/Heights Dataset	0.555556	0.684524	0.636765	0.545582
Blood Pressure Dataset	0.711443	0.750337	0.7215	0.705308
AskAPatient	0.629944	0.634494	0.63362	0.629799

Table 25 Pain SVM with SMOTENC

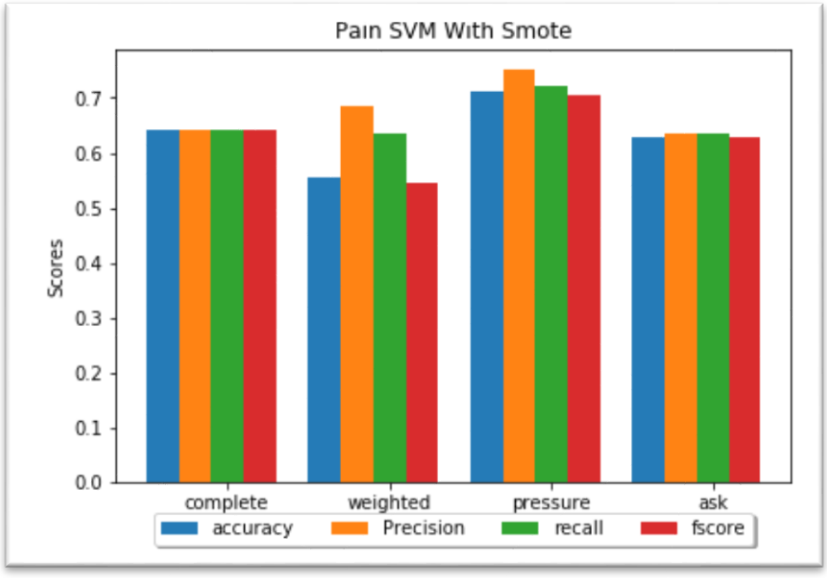


Figure 31 Pain SVM with SMOTENC

4.2.2.3. Naïve Bayes

Pain Prediction

- The results here have shown a slight improvement over SVM in pressure results, otherwise all other results are mediocre.

Dataset	Accuracy	Precision	Recall	FScore
Complete Dataset	0.553495	0.553395	0.553513	0.5532
Weights/Heights Dataset	0.703704	0.720833	0.733824	0.702069
Blood Pressure Dataset	0.626866	0.626733	0.62711	0.626533
AskAPatient	0.5	0.535696	0.520443	0.451768

Table 26 Pain NB

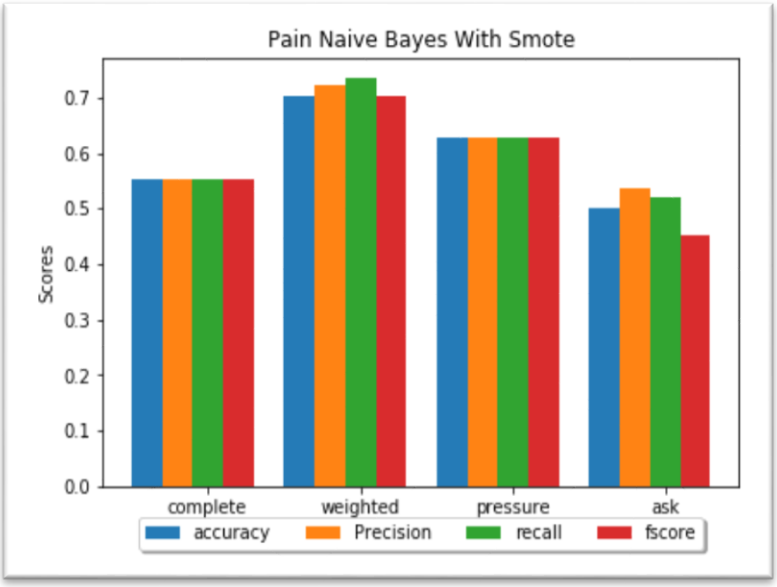


Figure 32 Pain NB

4.2.3. Second Experiment

- Predict the drug used based on the user profile and user history.
- Predict the drug family based on the user profile and user history.
- The user history is represented by ADR mention features.
- Testing the best number of estimators in terms of F-Score for random forests was made by trying all estimators between 100 and 1000 at 100 increment.

Labels/Datasets	Drugs	Drug Family
Complete	'Atenolol': 620, 'Lisinopril': 428, 'Amlodipine': 161, 'Hydrochlorothiazide': 129, 'Nadolol': 119, 'Diltiazem': 100	'Beta': 739, 'Angiotensin': 428, 'Calcium': 261, 'Diuretics': 129
Weight/Height	'Lisinopril': 51, 'Atenolol': 45, 'Amlodipine': 14, 'Hydrochlorothiazide': 13	'Angiotensin': 51, 'Beta': 45, 'Calcium': 14, 'Diuretics': 13
Blood Pressure	'Atenolol': 186, 'Lisinopril': 134, 'Amlodipine': 63, 'Hydrochlorothiazide': 51, 'Diltiazem': 26	'Beta': 186, 'Angiotensin': 134, 'Calcium': 89, 'Diuretics': 51
Ask A Patient	'Lisinopril': 622, 'Atenolol': 134	Angiotensin': 622, 'Beta': 134

Table 27 Dataset description 2

Drug Prediction

- AskAPatient had the highest result, which reflects its accuracy regarding ADR, disease and mental issues detection, since the mentions are clearer and easy to find, therefore accurately finding the user history is much easier than the rest.
- Weights/Highest and blood pressure are very close in results, both of which have better predictions than the complete dataset, which shows how much role these two information play in the predictions. Which might indicate that high blood pressure, extremely light or heavy weight, might cause diseases that lead to the use of drugs that in turn cause ADRs for using the drug with such extreme
- It should also be noted that AskAPatient is only limited to two drugs, which might explain its high results, as it is easier to identify than the rest.

Dataset	Accuracy	Precision	Recall	FScore
Complete Dataset	0.53405	0.530371	0.535376	0.532069
Weights/Heights Dataset	0.629032	0.614183	0.629097	0.617931
Blood Pressure Dataset	0.620072	0.614306	0.624035	0.616403
AskAPatient	0.802139	0.803151	0.802735	0.802116

Table 28 Drug Prediction Random Forests

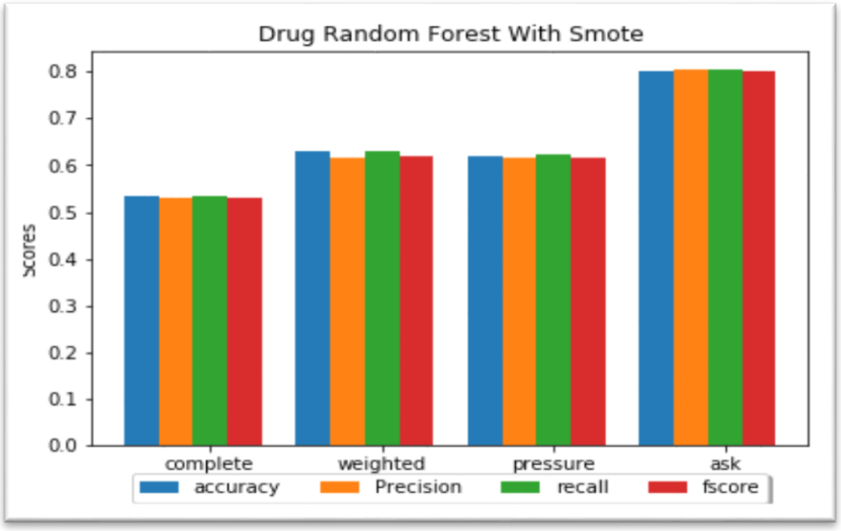


Figure 33 Drug Predictions Random Forests

Drug Family Predictions

- AskAPatient resutls did not change at all due to both drugs being from only two drug families, meaning that the exact same classification is made, just under different name.
- The Complete MedHelp dataset had better performance identifying the drug family, mainly because the number of distinct labels have decreased to a more manageable four classes instead of 6.
- As for the decrease in weights/heights and pressure performance, there is no clear explanation for weights/heights since the same conditions as AskAPatient apply here, pressure on the other hand could be explained by possible difference of the two calcium-blockers have in common and their different uses, which results in poorer results when their drug family is identified.

Dataset	Accuracy	Precision	Recall	FScore
Complete Dataset	0.572717	0.573601	0.574943	0.569366
Weights/Heights Dataset	0.596774	0.593651	0.599063	0.596121
Blood Pressure Dataset	0.544643	0.559338	0.550519	0.545089
AskAPatient	0.802139	0.803151	0.802735	0.802116

Table 29 Drug Family Prediction Random Forest

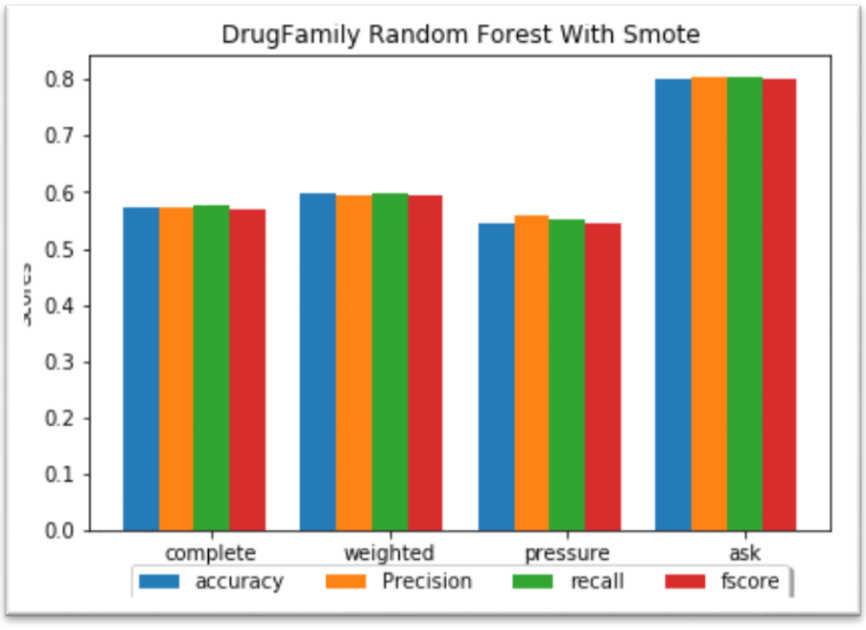


Figure 34 Drug Family Prediction Random Forest

4.2.4. Third Experiment

- This experiment compares the result between the first experiment's limit on using user profile data only, and now using all other user history data in addition to the profile. The chosen labels are a part of the history, which are removed when the experiment runs.
- Pain will be used as the main label for this experiment, since it is the most common ADR, having the same values as the first experiment.

Results:

The usage of user medical history has proven to be a success, being able to predict the appearance of pain with more than 10% better

Dataset	Accuracy	Precision	Recall	FScore
Complete Dataset	0.828816	0.836927	0.832474	0.828562
Weights/Heights Dataset	0.925926	0.916193	0.930882	0.922078
Blood Pressure Dataset	0.880597	0.880785	0.880105	0.880357
AskAPatient	0.79661	0.799699	0.799699	0.79661

Table 30 Pain prediction random forests

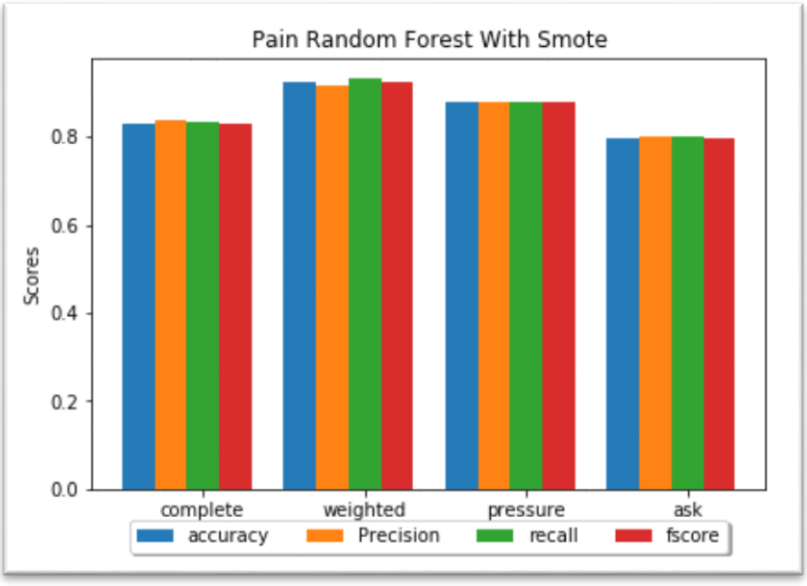


Figure 35 Pain prediction random forests

Chapter 5: Conclusion

5.1. Conclusion

- A web crawler successfully extracted a wide dataset from specialized medical forums.
- The dataset acquired successfully tested the capabilities of GloVe to find related concepts that, which helped in identifying the most frequent ADR in relation to a drug.
- Sliding window pattern detection was used to extract the user profile data- mainly age, gender, weight, height and blood pressure- to be used as features for various machine learning algorithms.
- A dictionary following the UMLS definitions was built using text with the help of MetaMap, and parsing its output allowed easy translation from plain text to python dictionary, the UMLS concepts- ADRs, diseases and mental issues- were used as features and classes for the supervised learning algorithm.
- Association rule mining using Apriori was tested to find any relationship between drug, drug family, gender of user, diseases and ADRs. It failed in acquiring any useful information within the tight constraints placed, however it was successful as a proof of concept.
- A comparison between the datasets before and after applying SMOTENC was made in the first experiment. The results provided show that not applying SMOTENC leads to higher accuracy, but had a smaller values for the other more important metrics, namely Recall, Precision and F1-Score (FScore).
- Random Forests outperformed both SVM and Naïve Bayes, both when using SMOTENC and without, and even when using feature filtering, therefore it was decided to ignore it on the third experiment if it did not produce a better result, which it did not.
- Feature filtering using random forests in general produced varying results, both positive and negative in all metrics, but generally there was not a definitive improvement over not using it. While reviewing the second experiment, it appears that the count of diseases, and mental issues were a major factor in improving the results in some cases, and that it was chosen as the most important besides user age.

- Feature filtering did not choose ADR count as important, and after trying to find it in the first experiment the results were extremely terrible in most the performance metrics, the same results were found in disease and mental issues count.
- The first and third experiments tried to predict the presence of the ADRs and diseases based on user profile, at 66% F-Score for Pain in MedHelp, the first experiment was successful with good results, however when using medical history from other ADRs and disease mentions in the third, this result has improved to 82% .
- As a comparison with some related work, this prediction was made to predict if the user got afflicted by any ADR, no specifics given other than the presence, this has proven to be successful and comparable with some results given in some of the papers on supervised learning, more specifically the result in figure 2 and figure 3, even the results without applying SMOTE were satisfactory, and they show the potential of user profile and medical history as features, given that more bigger and more diverse dataset is provided.

5.2. Future Work

- A bigger and wider range dataset, with more specialized UMLS concepts and a wider range of drugs, could be used for more advanced pharmacovigilance techniques.
- The crawler designed with JSoup could be modified to extract other data from various medical forums to provided easier access to data for medical research.
- The use of user profile as indication for medical problems could be further developed for a recommendation system, where any person with a certain medical history and have certain attributes could be recommended the use of certain drug that is not known to produce any severe ADRs for such attributes and history.

References

[1] R. Edwards and F. K. Aronson, "Adverse Drug Reactions: Definitions, Diagnosis and Management," *Lancet,* vol. 356, no. Number 9237, pp. 1255-1259, 7 October 2000.

[2] World Health Organization, "The Importance of Pharmacovigilance-Safety Monitoring of Medicinal Products," 2002. [Online]. Available: http://apps.who.int/medicinedocs/en/d/Js4893e/1.html. [Accessed 18 11 2018].

[3] A. Nikfrajam, K. Oconnor, A. Sarker, R. Ginn, S. Jayaraman, T. Upadhya and G. Gonzalez, "Utilizing social media data for pharmacovigilance: A review," *Research Gate,* vol. 54, pp. 202-212, 2015.

[4] US Food & Drug Administration (FDA), "Preventable Adverse Drug Reactions: A Focus on Drug Interactions," US Food & Drug Administration (FDA), 6 3 2018. [Online]. Available: https://www.fda.gov/drugs/developmentapprovalprocess/developmentresources/druginteractionslabeling/ucm110632.htm. [Accessed 19 11 2018].

[5] R. Leaman, L. Wojtulewicz, R. Sullivan, A. Shariah, J. Yang and G. Gonzalez, "Towards Internet-Age Pharmacovigilance: Extracting Adverse Drug," in *Proceedings of the 2010 workshop On biomedical Natural Language Processing*, Uppsala, Sweden , 2010.

[6] A. Nikfarjam and G. H. Gonzalez, "Pattern Mining for Extraction of mentions of Adverse Drug Reactions From User Comments," *AMIA Annual Symposium Proceedings Archive,* vol. 2011, pp. 1019-1026, 2011.

[7] D. Pappel, "Up To 90 Percent Of Adverse Reactions Unreported, Pharma Companies Search Social Media To Make Drugs Safer," Inquisitr, 14 3 2015. [Online]. Available: https://www.inquisitr.com/1914900/up-to-90-percent-of-adverse-reactions-unreported-pharma-companies-search-social-media-to-make-drugs-safer/. [Accessed 19 11 2018].

[8] R. Sloane, O. Osanlou, D. Lewis, D. Bollegala, S. Maskell and M. Pirmohamed, "Social media and pharmacovigilance: A review of the opportunities and challenges," *British Journal of Clinicla Pharmacology,* vol. 80, no. 4, pp. 910-920, 2015.

[9] C. C. Yang and H. Yang, "Exploiting Social Media With Tensor Decomposition For Pharmacovigilance," in *2015 IEEE 15th International Conference on Data Mining Workshops*, Atlantic City New Jersey, 2015.

[10] J. Hadzi-Puric and J. Grumsa, "Automatic Drug Adverse Reaction Discovery From Parenting websites using disproportionality methods," in *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, Istanbul, 2012.

[11] A. Sarker and G. Graciela, "Portable Automatic Text Classification for Adverse Drug Reaction Detection Via Multi-Corpus Training," *Journal of biomedical informatics*, vol. 53, pp. 196-207, 2014.

[12] J. Bian, U. Topaloglu and F. Yu, "Towards Large-Scale Twitter Mining For Drug-Related Adverse Events," in *Proceedings of the 2012 ACM International Workshop on Smart Health and Wellbeing*, Hawaii, 2012.

[13] Y. Zheng and K. Jiang, "Mining Twitter Data for Potential Drug Effects," in *Advanced Data Mining and Applications*, Hangzhou, Springer, 2013, pp. 434-443.

[14] A. Nikfarjam, A. Sarker, K. OConnor, R. Ginn and G. Gonzalez, "Pharmacovigilance from social media: mining adverse drug reaction mentions using sequence labeling with word embedding cluster features," *J Am Med Inform Assoc*, vol. 22, no. 3, pp. 671-681, 2015.

[15] H. Sampathkumar, B. Luo and X. W. Chen, "Mining Adverse Drug Reactions from online healthcare forums using Hidden Markov Model," *BMC Medical informatics and Decision Making*, vol. 14, no. 91, 2014.

[16] M. Yang, X. Wang and M. Kiang, "IDENTIFICATION OF CONSUMER ADVERSE DRUG REACTION MESSAGES ON SOCIAL MEDIA," in *PACIS 2013*, 2013.

[17] J. Brownlee, "Classification Accuracy is Not Enough: More Performance Measures You Can Use," Machine Learning Mastery, 21 3 2014. [Online]. Available: https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/. [Accessed 25 11 2018].

[18] Google, "Classification: Precision and Recall," Google, 1 10 2018. [Online]. Available: https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall. [Accessed 25 11 2018].

[19] W. Koehrsen, "Beyond Accuracy: Precision and Recall," Towards Data Science, 3 3 2018. [Online]. Available: https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c. [Accessed 25 11 2018].

[20] US Food & Drug Administration (FDA), "Questions and Answers on FDA's Adverse Event Reporting System (FAERS)," US Food & Drug Administration (FDA), 4 6 2018. [Online]. Available: https://www.fda.gov/Drugs/GuidanceComplianceRegulatoryInformation/Surveillance/AdverseDrugEffects/. [Accessed 20 11 2018].

[21] US Food & Drug Administration (FDA), "MedWatch: The FDA Safety Information and Adverse Event Reporting Program," US Food & Drug Administration (FDA), 14 11 2018. [Online]. Available: https://www.fda.gov/Safety/MedWatch/default.htm. [Accessed 20 11 2018].

[22] S. Fox, "The Social Life of Health Information," 12 5 2011. [Online]. Available: http://www.pewinternet.org/2011/05/12/the-social-life-of-health-information-2011/. [Accessed 20 11 2018].

[23] S. Vilar, C. Friedman and G. Hripcsak, "Detection of drug–drug interactions through data mining studies using clinical sources, scientific literature and social media," *Briefings in Bioinformatics*, vol. 19, no. 5, p. 863–877, 28 9 2018.

[24] Arizona State University, "TWITTER ANNOTATED CORPUS," Arizona State University, 2014. [Online]. Available: http://diego.asu.edu/downloads/twitter_annotated_corpus/. [Accessed 20 11 2018].

[25] DailyStrength, "DailyStrength: Getting Better Together," DailyStrength, 2006. [Online]. Available: https://www.dailystrength.org/. [Accessed 20 11 2018].

[26] MedHelp, "MedHelp: Be your healthiest," MedHelp, [Online]. Available: https://www.medhelp.org/. [Accessed 20 11 2018].

[27] PatientsLikeMe, "PatientsLikeMe: Living better starts here.," PatientsLikeMe, 2005. [Online]. Available: https://www.patientslikeme.com/. [Accessed 20 11 2018].

[28] Medications.com, "The Premier Community to talk about Health," [Online]. Available: http://www.medications.com/. [Accessed 2 12 2018].

[29] Ask A patient, "When you need to know if a medication really works, why not Ask a Patient?," [Online]. Available: https://www.askapatient.com/. [Accessed 2 12 2018].

[30] E. K. Ikonomakis, T. V. and K. Sotiris, "Text Classification Using Machine Learning," *WSEAS Transactions on Computers,* vol. 4, no. 8, pp. 966-974, 2005.

[31] U.S. National Library of Medicine, "Metathesaurus," U.S. National Library of Medicine, 12 4 2016. [Online]. Available: https://www.nlm.nih.gov/research/umls/knowledge_sources/metathesaurus/. [Accessed 28 11 2018].

[32] SiderEffects, "SIDER 4.1 : Side Effect Resource," SiderEffects, 21 10 2015. [Online]. Available: http://sideeffects.embl.de/. [Accessed 28 11 2018].

[33] Government of Canadian, "MedEffect Canada," Government of Canadian, 26 11 2018. [Online]. Available: https://www.canada.ca/en/health-canada/services/drugs-health-products/medeffect-canada.html. [Accessed 28 11 2018].

[34] Tartarus, "Snowball," Tartarus, 2 2002. [Online]. Available: http://snowball.tartarus.org/. [Accessed 28 11 2018].

[35] Rosettacode, "Jaro distance," Rosettacode, 29 7 2018. [Online]. Available: https://rosettacode.org/wiki/Jaro_distance. [Accessed 28 11 2018].

[36] B. Liu, W. Hsu and Y. Ma, "Integrating Classification and Association Rule Mining," in *The Fourth International Conference on Knowledge Discovery and Data Mining.*, New York, 1998.

[37] The Stanford NLP Group, "Software > Stanford Log-linear Part-Of-Speech Tagger," The Stanford NLP Group, 16 10 2018. [Online]. Available: https://nlp.stanford.edu/software/tagger.shtml. [Accessed 28 11 2018].

[38] Princeton University, "WordNet A Lexical Database for English," Princeton University, 2018. [Online]. Available: https://wordnet.princeton.edu/. [Accessed 2018].

[39] Christian Borgelt's Web Pages, "Apriori - Association Rule Induction / Frequent Item Set Mining," Christian Borgelt's Web Pages, 18 10 2018. [Online]. Available: http://www.borgelt.net/apriori.html. [Accessed 28 11 2018].

[40] A. C. Egberts, R. H. Meyboom and E. P. van Puijenbroek, "Use of Measures of Disproportionality in Pharmacovigilance," *Spriner Link*, vol. 25, no. 6, pp. 453-458, 2002.

[41] Serbian Government, "Medecines and Medical Devices Agency of Serbia," Serbian Government, [Online]. Available: https://www.alims.gov.rs/eng/. [Accessed 2 12 2018].

[42] DrugBank, "The DrugBank database is a unique bioinformatics and cheminformatics resource that combines detailed drug data with comprehensive drug target information.," DrugBank, [Online]. Available: https://www.drugbank.ca/. [Accessed 2 12 2018].

[43] USA government, "MetaMap - A Tool For Recognizing UMLS Concepts in Text," USA government, 23 1 2018. [Online]. Available: https://metamap.nlm.nih.gov/. [Accessed 29 11 2018].

[44] B. W. Chee, R. Berlin and B. Schatz, "Predicting Adverse Drug Events from Personal Health Messages," in *AMIA Annual Symposium Proceedings Archive*, Washington DC, 2011.

[45] MedDRA, "Welcome to MedDRA," MedDRA, [Online]. Available: https://www.meddra.org/. [Accessed 1 12 2018].

[46] Brandeis University, "Inter Annotator Agreement," Word Press, 28 2 2017. [Online]. Available: https://corpuslinguisticmethods.wordpress.com/2014/01/15/what-is-inter-annotator-agreement/. [Accessed 29 11 2018].

[47] Sphinx, "Natural Language Toolkit," Sphinx, 17 11 2018. [Online]. Available: http://www.nltk.org/. [Accessed 29 11 2018].

[48] Carnegie Mellon, "Tweet NLP," Carnegie Mellon, [Online]. Available: http://www.cs.cmu.edu/~ark/TweetNLP/. [Accessed 29 11 2018].

[49] University of Massachusetts Amherst, "MAchine Learning for LanguagE Toolkit," University of Massachusetts Amherst, 2002. [Online]. Available: http://mallet.cs.umass.edu/index.php. [Accessed 29 11 2018].

[50] B. O'Connor, R. Balasubramanyan, B. R. Routledge and N. A. Smith, "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series," in *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*, 2010.

[51] M. J. Paul and M. Dredze, "You are what you Tweet: Analysing Twitter fo public Health," in *file:///C:/Users/Ahmed/Downloads/2880-14200-1-PB.pdf*, Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, 2011.

[52] Apache, "Apache Lucene Core," Apache, 2011. [Online]. Available: https://lucene.apache.org/core/. [Accessed 1 12 2018].

[53] N. Okazaki, "CRFsuite A fast implementation of Conditional Random Fields (CRFs)," Naoaki Okazaki's website, 25 5 2016. [Online]. Available: http://www.chokkan.org/software/crfsuite/. [Accessed 2 12 2018].

[54] Google, "word2vec," Apache, 30 7 2013. [Online]. Available: https://code.google.com/archive/p/word2vec/. [Accessed 2 12 2018].

[55] SteadyHealth, SteadyHealth, [Online]. Available: https://www.steadyhealth.com/. [Accessed 2 12 2018].

[56] Drugs.com, "Worried about drug interactions? Use the Interactions Checker," Drugs.com, [Online]. Available: https://www.drugs.com/. [Accessed 2 12 2018].

[57] L. Zhang, S. Wang and B. Liu, "Deep learning for sentiment analysis:A survey," 14 2 2018. [Online]. Available: https://onlinelibrary.wiley.com/doi/epdf/10.1002/widm.1253. [Accessed 9 6 2019].

[58] NSS, "An Intuitive Understanding of Word Embeddings: From Count Vectors to Word2Vec," Analytics Vidhya, 4 6 2017. [Online]. Available: https://www.analyticsvidhya.com/blog/2017/06/word-embeddings-count-word2veec/. [Accessed 9 6 2019].

[59] J. Pennington, "GloVe: Global Vectors for Word Representation," Stanford, 8 2014. [Online]. Available: https://nlp.stanford.edu/projects/glove/. [Accessed 10 4 2019].

[60] Towards DataScience, "20. GLoVe - Global Vectors for Word Representation," Towards DataScience, 13 1 2019. [Online]. Available: https://www.youtube.com/watch?v=Kc2IXCpdEoM. [Accessed 9 6 2019].

[61] Tensorflow, "Embeddings," Tensorflow, [Online]. Available: https://www.tensorflow.org/guide/embedding. [Accessed 8 5 2019].

[62] Tensorflow, "Embedding Projector," Tensorflow, [Online]. Available: https://projector.tensorflow.org/. [Accessed 8 5 2019].

[63] U. Malik, "Association Rule Mining via Apriori Algorithm in Python," Stack Abuse, 9 8 2018. [Online]. Available: https://stackabuse.com/association-rule-mining-via-apriori-algorithm-in-python/. [Accessed 8 6 2019].

[64] D. Bhende, U. Kosarker and M. Gedam, "Study of various Improved Apriori Algorithms," *Journal of Computer Engineering* , vol. 13, pp. 55-58, 2016.

[65] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research* , vol. 16, pp. 321-357, 2002.

[66] imbalanced-learn, "imblearn.over_sampling.SMOTENC," imbalanced-learn, 2016. [Online]. Available: https://imbalanced-learn.readthedocs.io/en/stable/generated/imblearn.over_sampling.SMOTENC.html. [Accessed 3 6 2019].

[67] J. Khurkhuriya, "SMOTE - Synthetic Minority Oversampling Technique," Jitesh Khurkhuriya, 29 1 2018. [Online]. Available: https://www.youtube.com/watch?v=FheTDyCwRdE&t=306s. [Accessed 4 6 2019].

[68] A. Onan, S. Korukoglu and H. Bulut, "Ensemble of keyword extraction methods and classifiers in text Clasification," *Expert Systems With Applications*, vol. 57, pp. 232-247, 2016.

[69] L. Breiman, "RANDOM FORESTS," 2001. [Online]. Available: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf. [Accessed 5 6 2019].

[70] J. Stamer, "StatQuest: Random Forests Part 1 - Building, Using and Evaluating," StatQuest with Josh Starmer, 5 2 2018. [Online]. Available: https://www.youtube.com/watch?v=J4Wdy0Wc_xQ. [Accessed 5 6 2019].

[71] A. Dubey, "Feature Selection Using Random forest," Towards Data Science, 15 12 2018. [Online]. Available: https://towardsdatascience.com/feature-selection-using-random-forest-26d7b747597f. [Accessed 6 6 2019].

[72] D. . K. SRIVASTAVA and L. BHAMBHU, "DATA CLASSIFICATION USING SUPPORT VECTOR MACHINE," 2005. [Online]. Available: http://www.jatit.org/volumes/research-papers/Vol12No1/1Vol12No1.pdf. [Accessed 6 6 2019].

[73] S. TAHERI and M. MAMMADOV, "LEARNING THE NAIVE BAYES CLASSIFIER WITH OPTIMIZATION MODELS," *International Journal of Applied Mathematics and Computer Science.*, vol. 23, no. 4, pp. 787-795, 2013.

[74] SciKit learn, "API Reference," SciKit learn, 2019. [Online]. Available: https://scikit-learn.org/stable/modules/classes.html#module-sklearn.naive_bayes. [Accessed 7 6 2019].

[75] M. Sunasra, "Performance Metrics for Classification problems in Machine Learning," Medium Corporation, 11 11 2017. [Online]. Available: https://medium.com/thalus-ai/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b. [Accessed 9 6 2019].

[76] R. N. Fogoros, "Drugs Commonly Used to Treat High Blood Pressure," verywellhealth, 2 5 2019. [Online]. Available: https://www.verywellhealth.com/hypertension-drugs-1745989. [Accessed 11 6 2019].

[77] Mathematics, "Why does the SVM margin is 2‖w‖," Mathematics, 30 5 2015. [Online]. Available: https://math.stackexchange.com/questions/1305925/why-does-the-svm-margin-is-frac2-mathbfw. [Accessed 13 6 2019].

Appendix 1 – Figures of other results

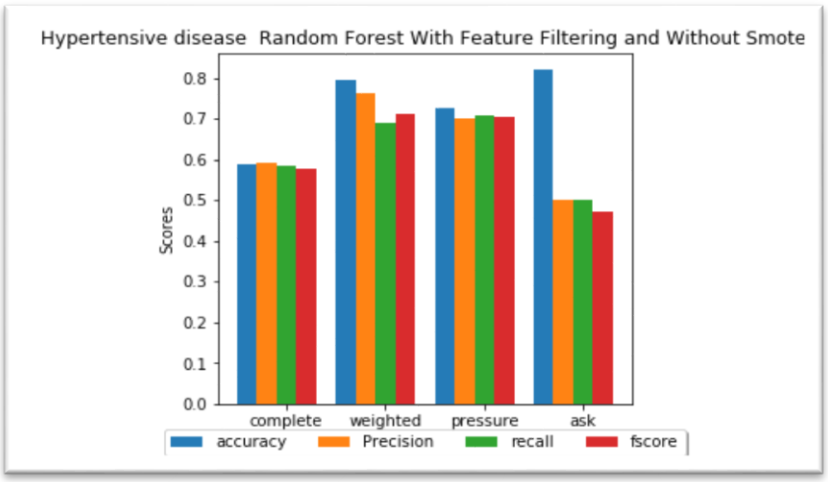
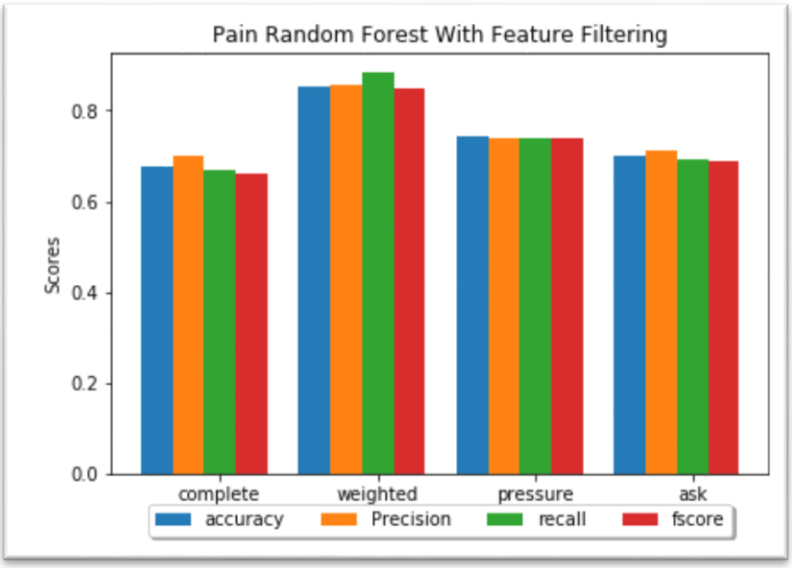


Figure 37 Hypertensive disease RF with Feature Filtering and without

Figure 38 Pain RF with Feature Filtering and SMOTENC



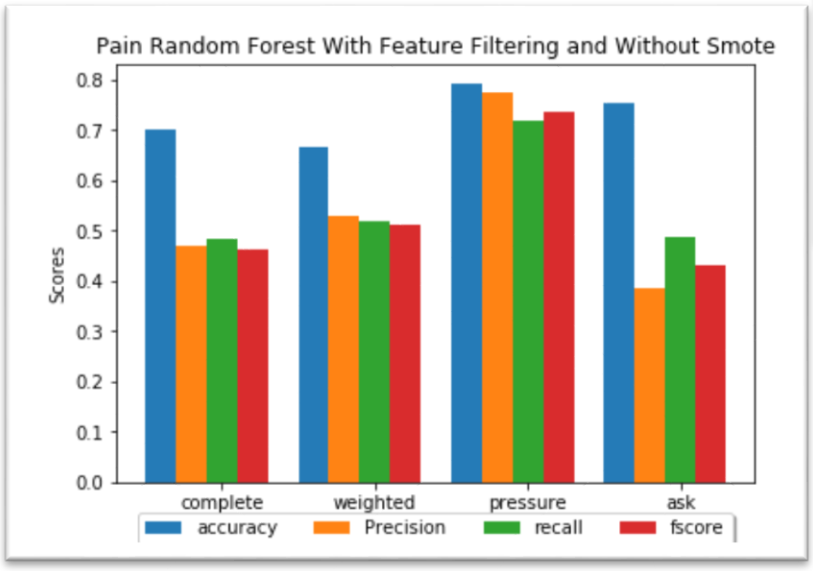


Figure 36 Pain RF with Feature Filtering and Without SMOTENC

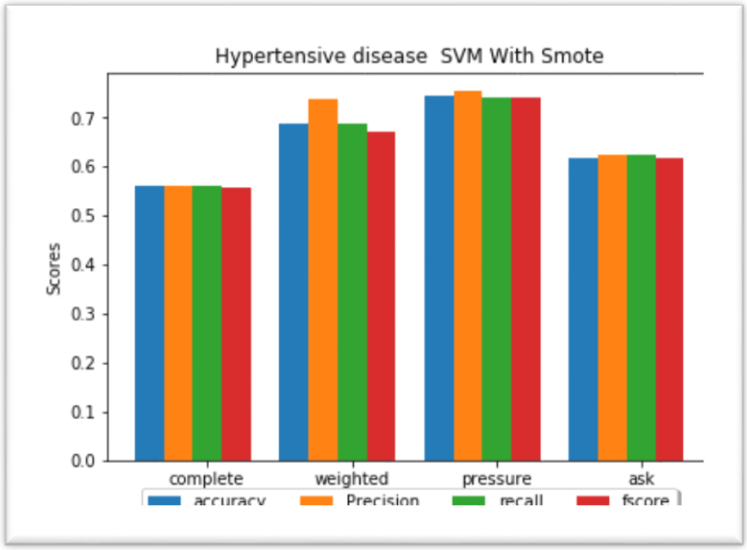


Figure 41 Hypertensive disease SVM with SMOTENC

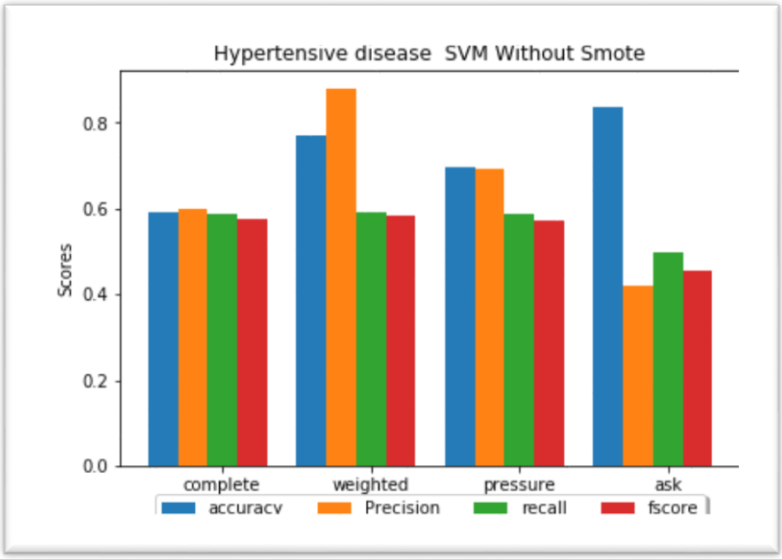


Figure 40 Hypertensive disease SVM without SMOTENC

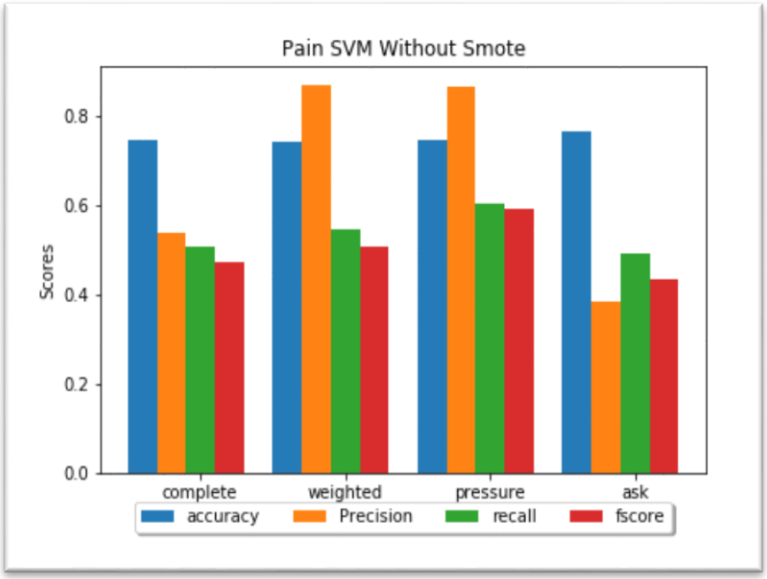


Figure 39 Pain SVM without SMOTENC

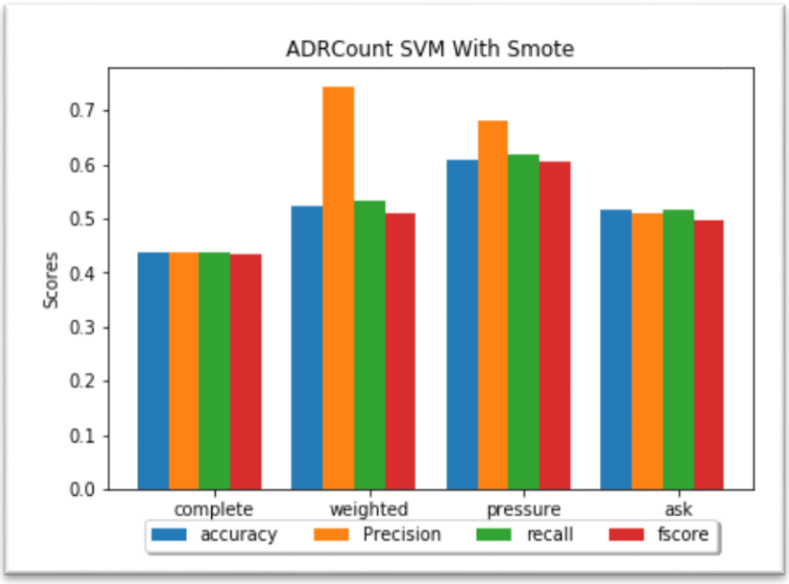


Figure 44 ADR Count with SMOTENC

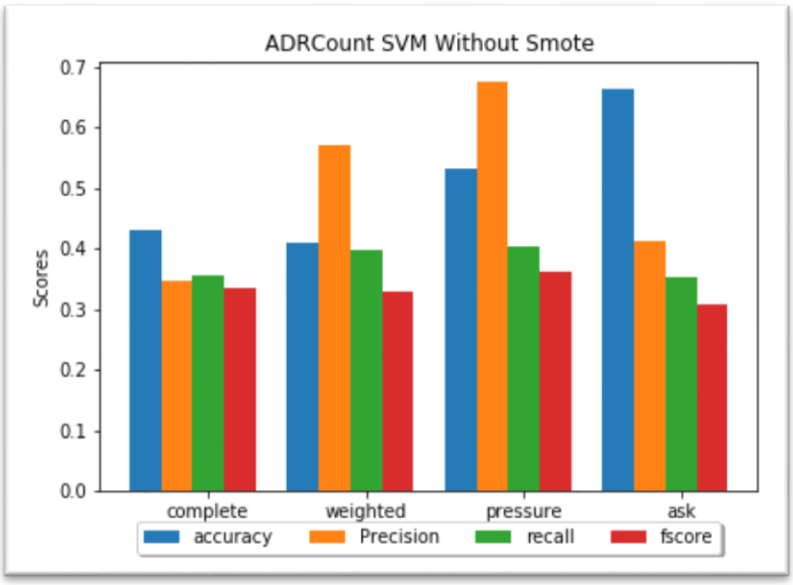


Figure 43 ADR Count Without SMOTENC

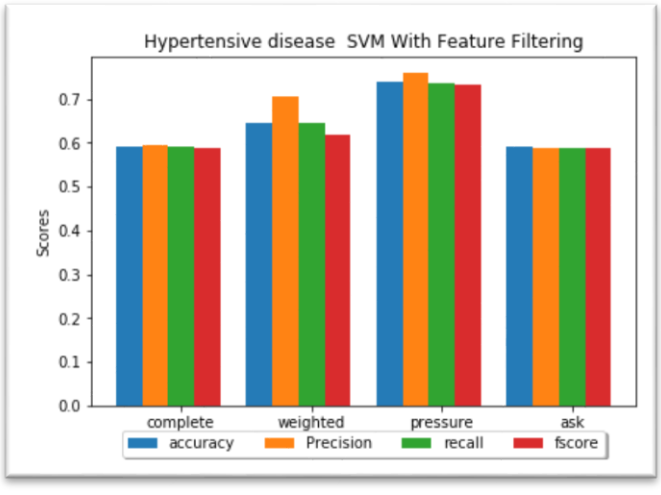


Figure 42 Hypertensive disease SVM with Feature Filtering

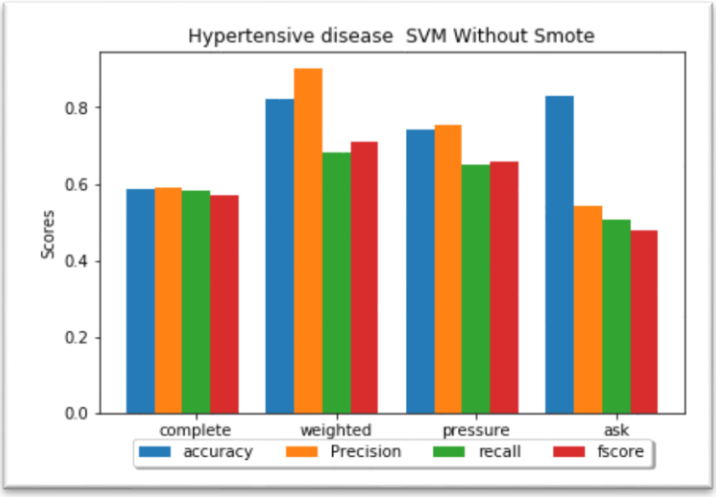


Figure 45 Hypertensive disease SVM with Feature Filtering without SMOTENC

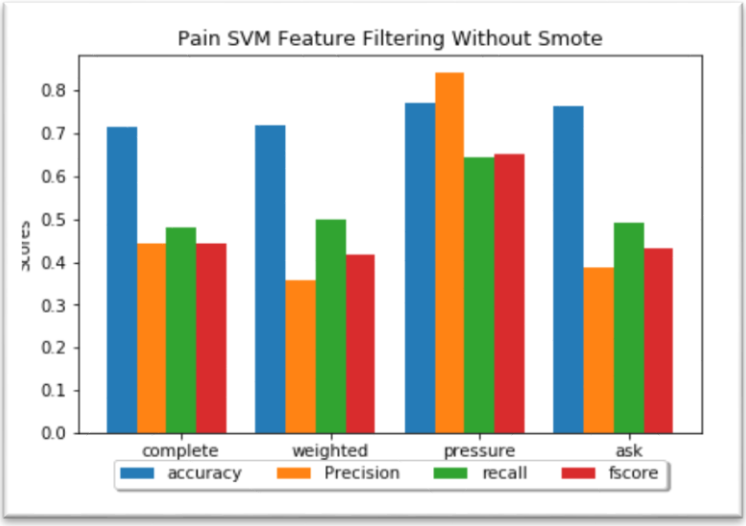


Figure 47 Pain SVM with Feature Filtering Without SMOTENC

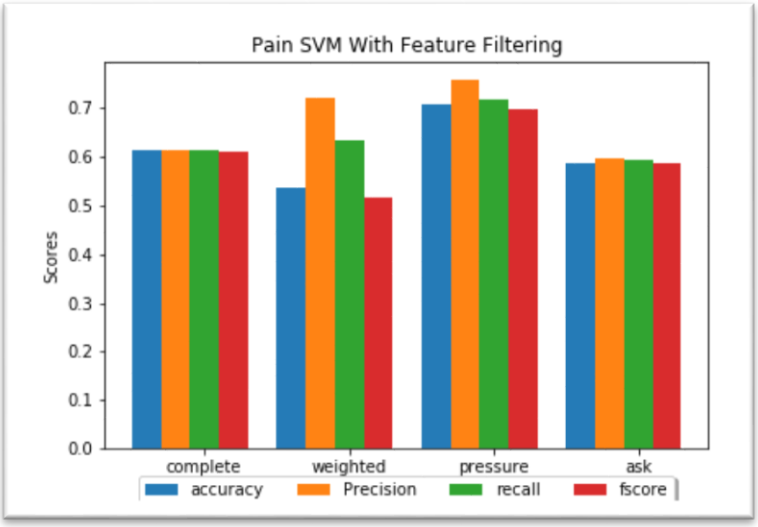


Figure 46 Pain SVM with Feature Filtering and SMOTENC

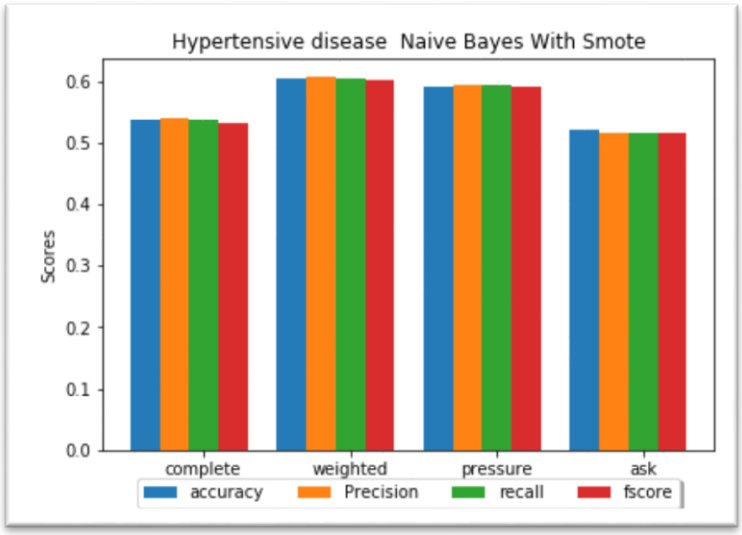


Figure 49 Hypertensive Disease NB

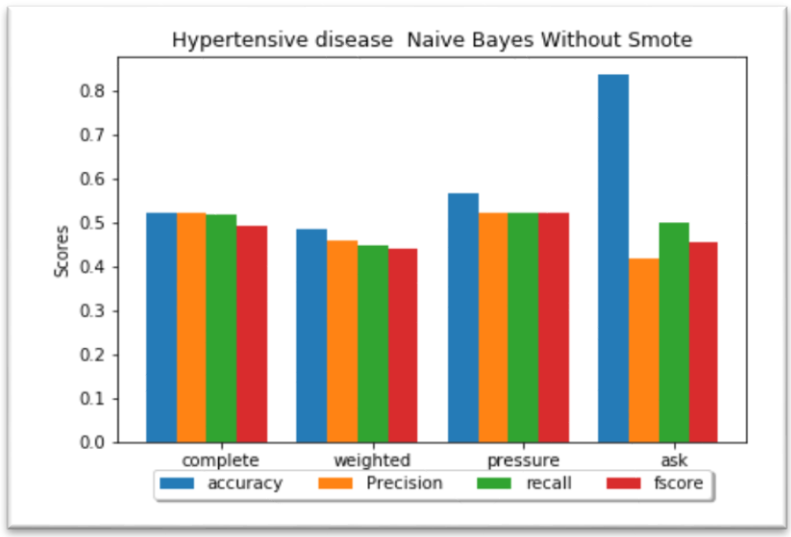


Figure 50 Hypertensive Disease NB without SMOTENC

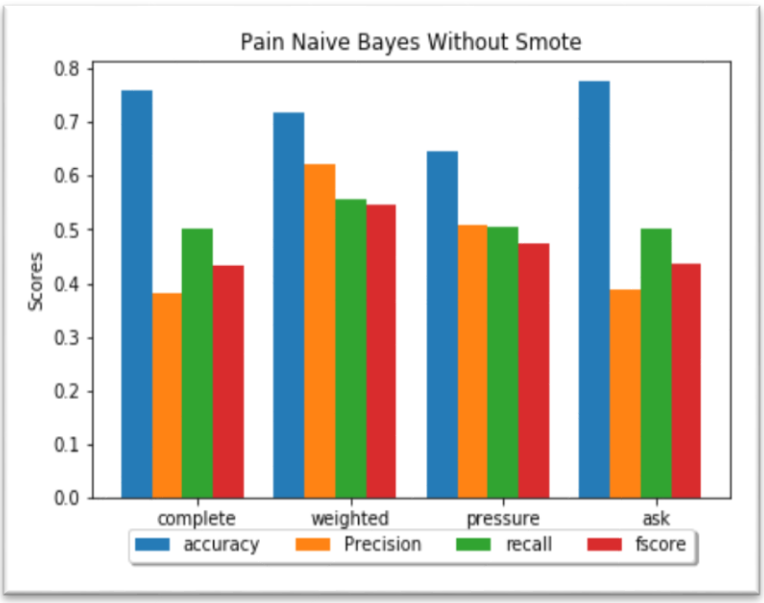


Figure 48 Pain NB Without Smote

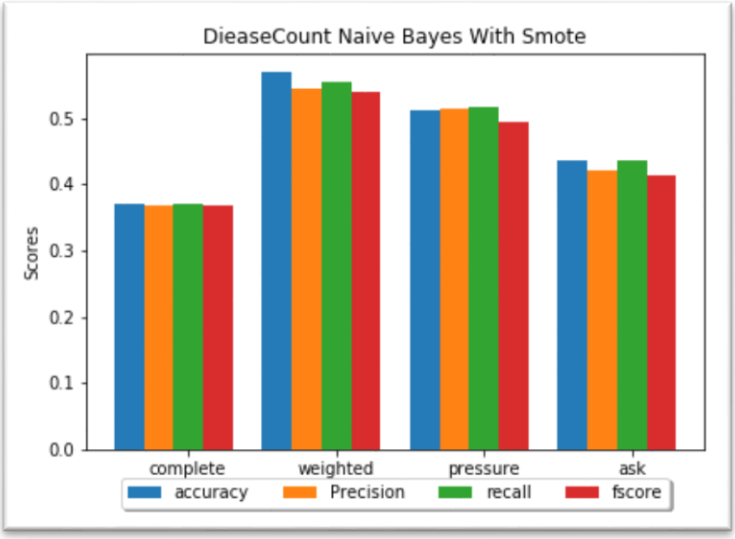


Figure 53 Disease Count NB

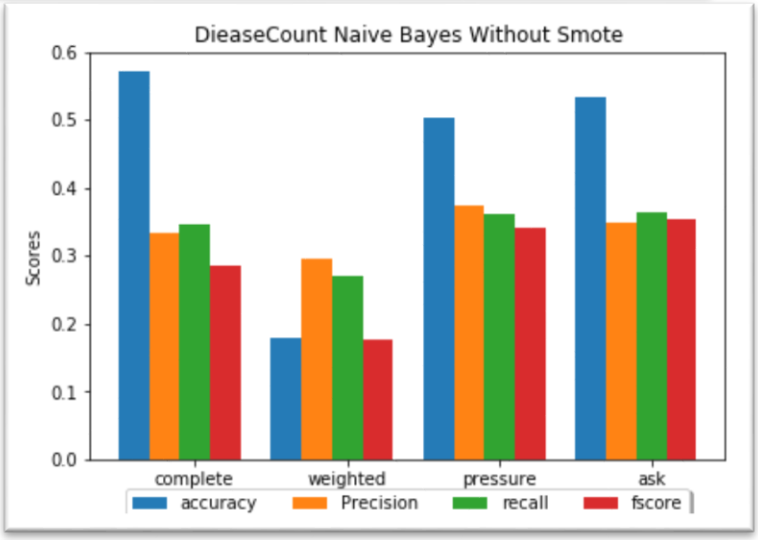


Figure 52 Disease Count NB without SMOTENC

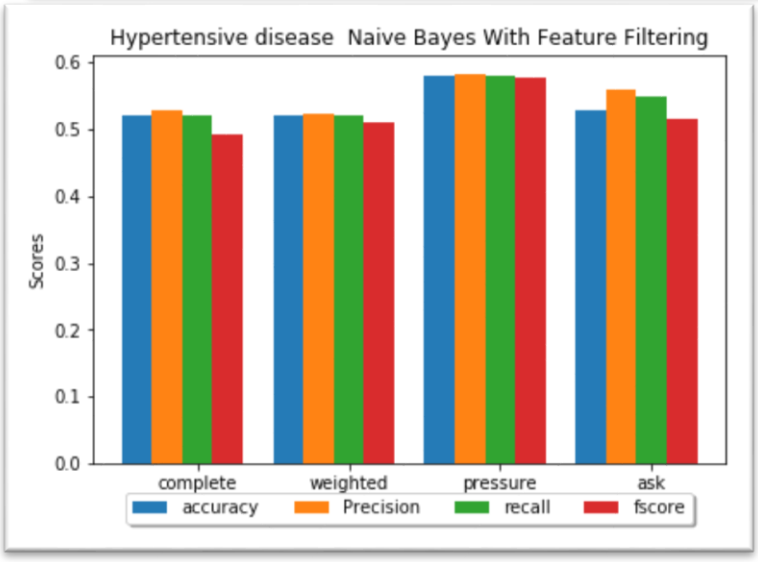


Figure 51 Hypertensive Disease NB with Feature Filtering

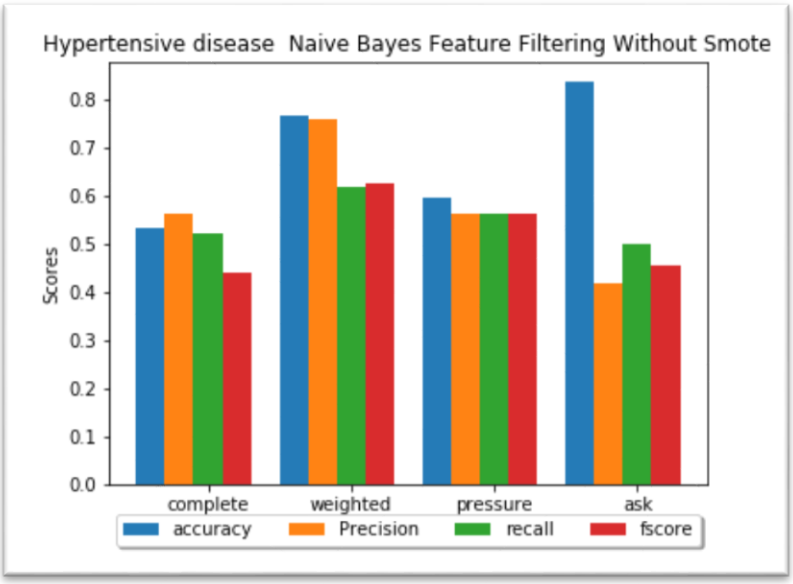


Figure 56 Hypertensive Disease NB with Feature Filtering without SMOTENC

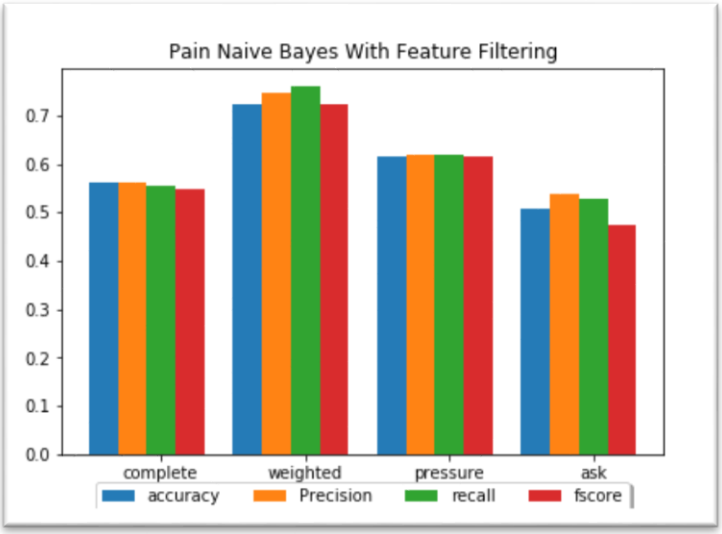


Figure 55 Pain NB with Feature Filtering

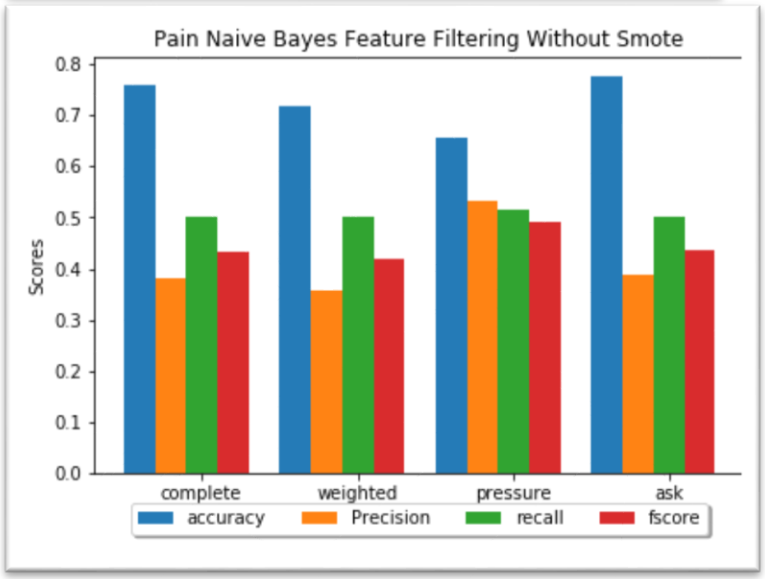


Figure 54 Pain NB with Feature Filtering without SMOTENC

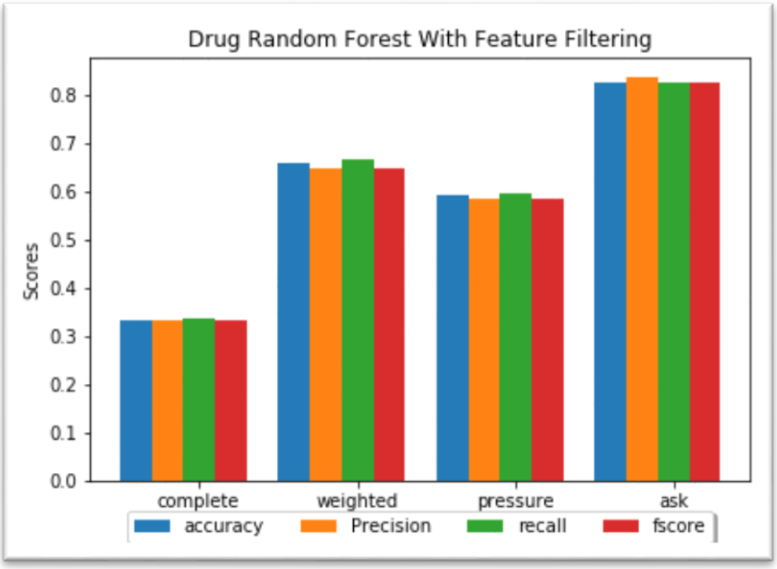


Figure 57 Drug Predictions Random Forest with Filtering

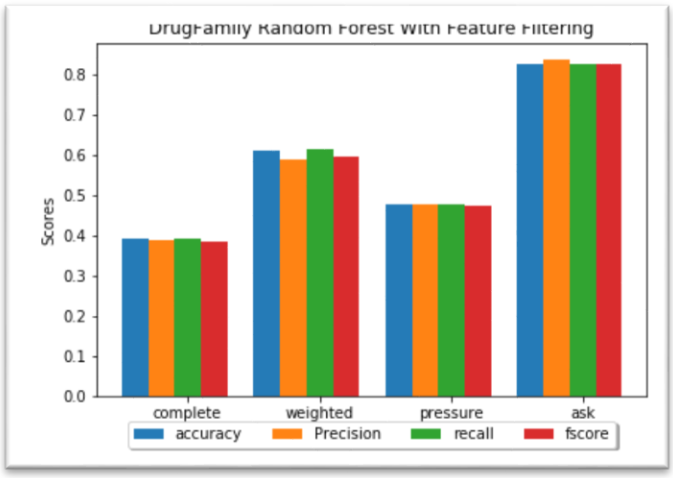


Figure 59 Drug Family predictions random Forest with feature filtering

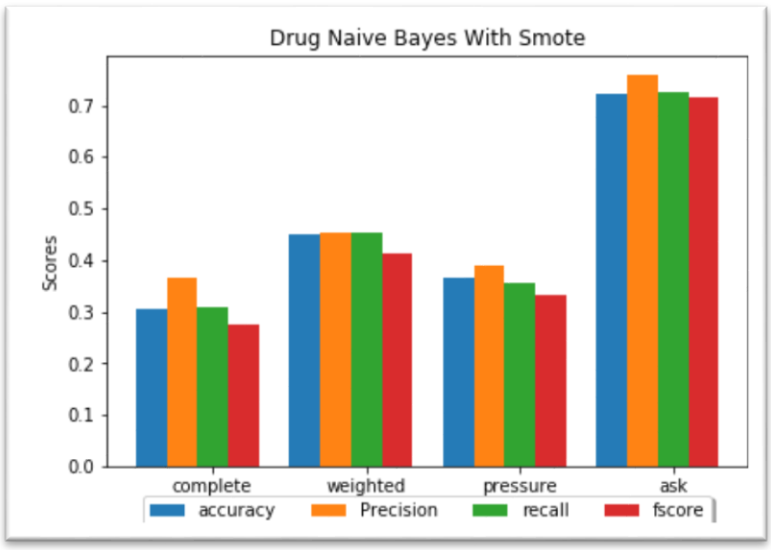


Figure 58 Drug Predictions with Naive Bayes

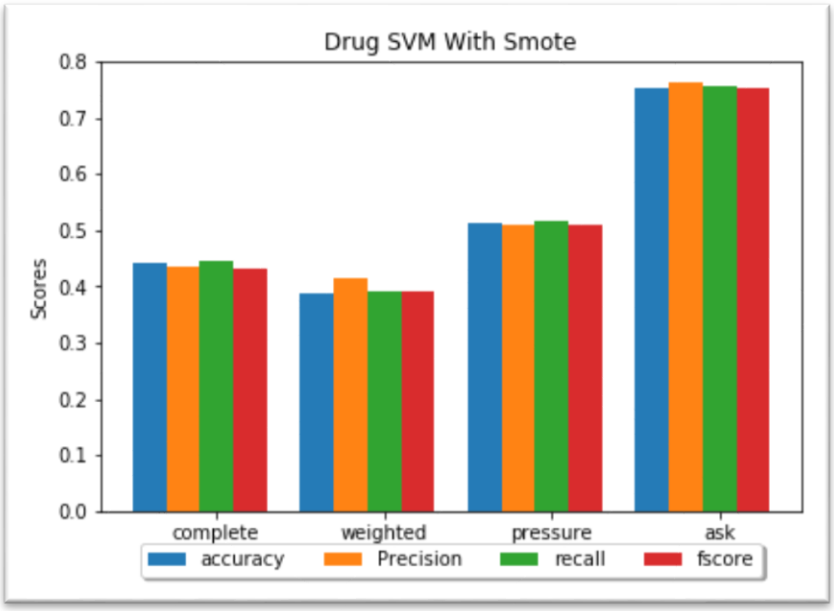


Figure 61 Drug Predictions SVM

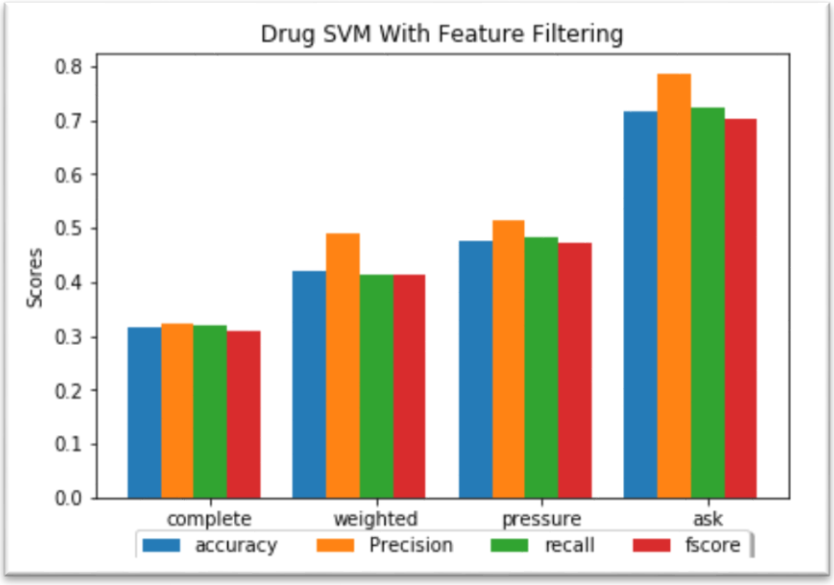


Figure 60 Drug prediction with SVM and Feature Filtering

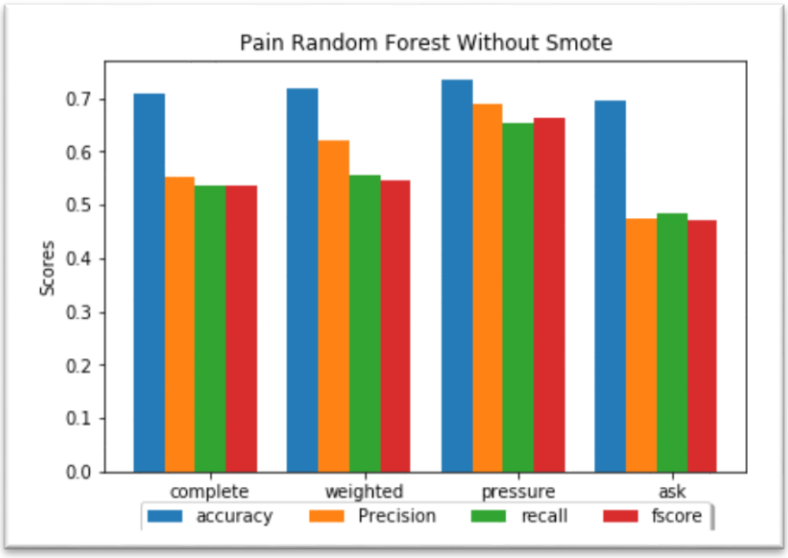


Figure 63 RF Pain without SMOTENC

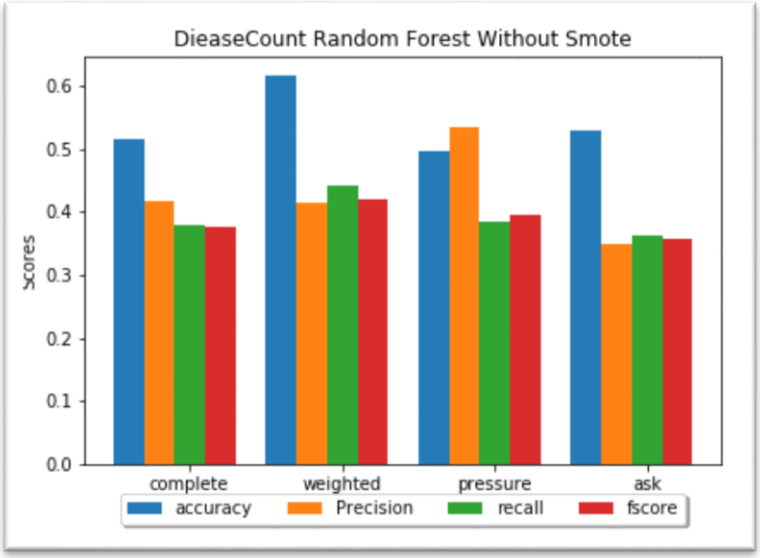


Figure 64 Disease Count RF without SMOTENC

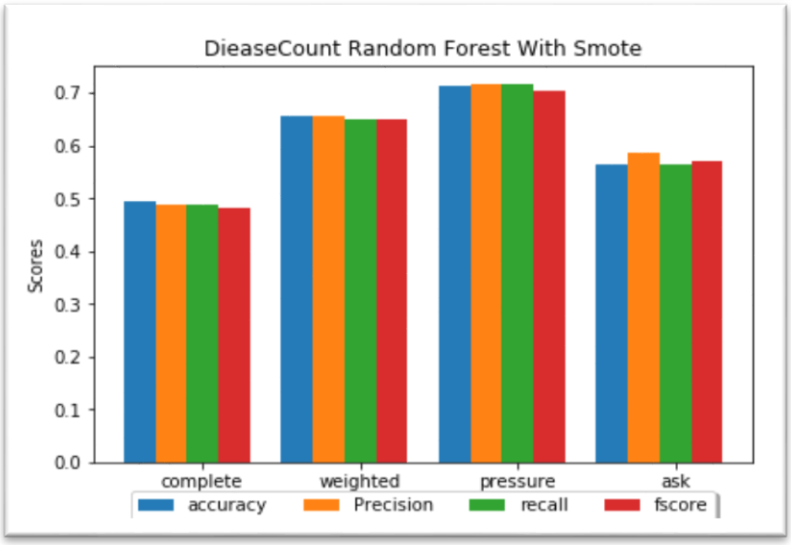


Figure 62 Disease Count RF with

Appendix 2 – Tables of other results

Datasets	Accuracy	Precision	Recall	FScore
Complete	0.709402	0.552477	0.536667	0.536188
Weighted	0.717949	0.621429	0.555195	0.546032
Pressure	0.733813	0.690711	0.652751	0.6622
Ask	0.696035	0.473708	0.483679	0.471291

Table 31 RF Pain without SMOTENC

Datasets	Accuracy	Precision	Recall	FScore
Complete	0.423077	0.39879	0.388964	0.391975
Weighted	0.333333	0.330159	0.330159	0.329885
Pressure	0.561151	0.499906	0.482774	0.479719
Ask	0.559471	0.382987	0.3419	0.336909

Table 32 RF DiseaseCount without SMOTENC

Datasets	Accuracy	Precision	Recall	FScore
Complete	0.514957	0.416226	0.377881	0.377375
Weighted	0.615385	0.415344	0.441414	0.42037
Pressure	0.496403	0.534444	0.384615	0.395995
Ask	0.528634	0.349699	0.363579	0.356326

Table 33 RF ADR Count without SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.587607	0.592206	0.583772	0.576133
Weighted	0.794872	0.763393	0.691558	0.711111
Pressure	0.726619	0.700376	0.707532	0.703258
Ask	0.819383	0.501885	0.500356	0.473377

Table 34 Hypertensive Disease RF with Featuer Filtering and without SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.677603	0.701309	0.668407	0.660118
Weighted	0.851852	0.857143	0.882353	0.85
Pressure	0.741294	0.740859	0.739424	0.739845
Ask	0.700565	0.711313	0.69165	0.689782

Table 35 Pain RF With Feature Filtering and SMOTENC

Estimators	Accuracy	Precision	Recall	FScore
Complete	0.587607	0.592206	0.583772	0.576133
Weighted	0.794872	0.763393	0.691558	0.711111
Pressure	0.726619	0.700376	0.707532	0.703258
Ask	0.819383	0.501885	0.500356	0.473377

Table 36 Pain RF with Feature Filtering and Without SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.560878	0.562123	0.561012	0.558965
Weighted	0.6875	0.737363	0.6875	0.670179
Pressure	0.743902	0.753535	0.742674	0.74078
AskAPatient	0.619178	0.6234	0.623864	0.619132

Table 37 Hypertensive disease SVM with SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.59188	0.599346	0.5875	0.577272
Weighted	0.769231	0.878378	0.590909	0.584615
Pressure	0.697842	0.693343	0.587111	0.573744
AskAPatient	0.837004	0.418502	0.5	0.455635

Table 38 Hypertensive disease SVM without SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.745726	0.539855	0.508527	0.471888
Weighted	0.74359	0.868421	0.545455	0.507576
Pressure	0.748201	0.865385	0.602273	0.592034
AskAPatient	0.76652	0.386667	0.494318	0.433915

Table 39 Pain SVM without SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.436102	0.435908	0.43593	0.43584
Weighted	0.522727	0.742424	0.531746	0.509794
Pressure	0.608696	0.680497	0.618276	0.604497
AskAPatient	0.515766	0.509676	0.515725	0.495939

Table 40 ADR Count with SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.431624	0.347332	0.35475	0.334189
Weighted	0.410256	0.571895	0.398413	0.328704
Pressure	0.532374	0.674731	0.404633	0.360942
AskAPatient	0.665198	0.413203	0.351358	0.308581

Table 41 ADR Count without SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.58547	0.591346	0.58125	0.571671
Weighted	0.820513	0.9	0.681818	0.711111
Pressure	0.741007	0.753885	0.649611	0.656319
AskAPatient	0.828194	0.544283	0.505619	0.477175

Table 42 Hypertensive disease SVM with Feature Filtering without SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.590818	0.593961	0.590996	0.587638
Weighted	0.645833	0.706388	0.645833	0.617799
Pressure	0.737805	0.758133	0.736055	0.731567
AskAPatient	0.591781	0.588914	0.589318	0.589004

Table 43 Hypertensive disease SVM with Feature Filtering

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.614836	0.613653	0.612231	0.612088
Weighted	0.537037	0.722222	0.632353	0.516995
Pressure	0.706468	0.756909	0.717875	0.697821
AskAPatient	0.587571	0.597529	0.594078	0.585666

Table 44 Pain SVM with Feature Filtering and SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.536926	0.539222	0.537155	0.530853
Weighted	0.604167	0.60582	0.604167	0.602614
Pressure	0.591463	0.594377	0.592369	0.589617
AskAPatient	0.520548	0.516347	0.516364	0.516351

Table 45 Hypertensive Disease NB

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.523504	0.522391	0.517544	0.492884
Weighted	0.487179	0.458556	0.449675	0.442857
Pressure	0.568345	0.522665	0.522665	0.522665
AskAPatient	0.837004	0.418502	0.5	0.455635

Table 46 Hypertensive Disease NB without SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.760684	0.380342	0.5	0.432039
Weighted	0.717949	0.621429	0.555195	0.546032
Blood	0.647482	0.509409	0.504187	0.472874
AskAPatient	0.77533	0.387665	0.5	0.436725

Table 47 Pain NB Without Smote

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.553495	0.553395	0.553513	0.5532
Weighted	0.703704	0.720833	0.733824	0.702069
Pressure	0.626866	0.626733	0.62711	0.626533
AskAPatient	0.5	0.535696	0.520443	0.451768

Table 48 Pain NB

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.371648	0.369142	0.369659	0.369044
Weighted	0.568966	0.544067	0.553872	0.539088
Blood	0.512821	0.513312	0.516115	0.494191
AskAPatient	0.435967	0.421373	0.435923	0.412535

Table 49 Disease Count NB

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.57265	0.371117	0.343385	0.271581
Weighted	0.538462	0.342857	0.360606	0.350697
Blood	0.546763	0.367236	0.359568	0.287233
AskAPatient	0.506608	0.173454	0.316804	0.224172

Table 50 NB Count without SMOTENC

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.520958	0.527749	0.521434	0.492315
Weighted	0.520833	0.52277	0.520833	0.510421
Blood	0.579268	0.581915	0.580173	0.577367
AskAPatient	0.528767	0.560603	0.549848	0.515227

Table 51 Hypertensive Disease NB with Feature Filtering

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.534188	0.565179	0.524013	0.439921
Weighted	0.769231	0.760714	0.618506	0.628571
Blood	0.597122	0.562113	0.564332	0.562697
AskAPatient	0.837004	0.418502	0.5	0.455635

Table 52 Hypertensive Disease NB with Feature Filtering without SMOTENC

Table 53 Pain NB with Feature Filtering

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.562054	0.560617	0.554822	0.547433
Weighted	0.722222	0.746844	0.758824	0.721362
Blood	0.616915	0.61735	0.617676	0.616764
AskAPatient	0.508475	0.538858	0.526307	0.474509

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.760684	0.380342	0.5	0.432039
Weighted	0.717949	0.358974	0.5	0.41791
Blood	0.654676	0.533028	0.51555	0.489908
AskAPatient	0.77533	0.387665	0.5	0.436725

Table 54 Pain NB with Feature Filtering without SMOTENC

Estimators	Accuracy	Precision	Recall	FScore
Complete	0.335125	0.334382	0.33591	0.333261
Weighted	0.66129	0.648049	0.668111	0.648689
Pressure	0.594982	0.586339	0.597276	0.585986
Ask	0.828877	0.836913	0.827197	0.827277

Table 55 Drug Predictions Random Forest with Filtering

Estimators	Accuracy	Precision	Recall	FScore
Complete	0.392334	0.391084	0.393215	0.385144
Weighted	0.596774	0.575284	0.597724	0.582342
Pressure	0.477679	0.47844	0.478799	0.476218
Ask	0.828877	0.836913	0.827197	0.827277

Table 56 Drug Family predictions random Forest with feature filtering

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.317204	0.324079	0.318835	0.308868
Weighted	0.419355	0.490196	0.413183	0.413526
Blood	0.476703	0.51323	0.481905	0.474449
AskAPatient	0.716578	0.784646	0.721712	0.701521

Table 57 Drug Predictions with SVM

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.443548	0.436758	0.445805	0.432164
Weighted	0.387097	0.414805	0.390319	0.390853
Pressure	0.512545	0.50988	0.515292	0.510708
AskAPatient	0.754011	0.761474	0.755729	0.752994

Table 59 Drug prediction with SVM and Feature Filtering

Dataset	Accuracy	Precision	Recall	FScore
Complete	0.305556	0.364391	0.310105	0.273662
Weighted	0.451613	0.454023	0.454578	0.414229
Blood	0.365591	0.389339	0.354946	0.333318
AskAPatient	0.721925	0.757815	0.725803	0.714067

Table 58 Drug Predictions with Naive Bayes