**Data Mining Project**

**Uber Sentiment Analysis**

Data Mining Project,

Dr MD KARIM,

KEITH MAULL,

Project report

**Group Members**:

ABHINAND SHARMA YAMAVARAM (0592)

AJAY GADDAM (0513)

KIRAN KUMAR ADEPU (0458)

**INDEX:**

1. Sentiment analysis
2. Project Overview
3. DATASET
4. Research Questions
5. Models and Analysis
6. Code and Application
7. Project Management
8. Team Management
9. Check Points And Deliverables
10. Final Output
11. Code
12. References

**UBER SENTIMENT ANALYSIS**

1. **Sentiment analysis:**

Sentiment analysis in other words defined as opinion mining. In essence, it is the process of determining the emotional tone behind a series of words, used to gain an understanding of the attitudes, opinions and emotions expressed within an online mention. This sentimental analysis is usually used by the business people or any web service providers to know the customers or people opinion on their work, a feedback from customers. This analysis will give us the positive or negative reviews on the customer opinions. For example we can consider Obama administration used the sentimental analysis to review the people opinion on the coming elections.

1. **Project Overview:**

This project mainly works on the sentiment analysis for UBER; this UBER is a transport service provider available almost in ten countries around the globe. This usually comes with an application available on all the smart phones on all the operating systems. This also works on the TFIDF scores i.e., term frequency–inverse document frequency. Now my project will classify the positive and negative feelings of customers using their reviews either posted on FACEBOOK or TWITTER. This mainly takes some reviews and trains the data according to the positive and negative analysis, after that this will take the keywords which will specify the positive and negative words and takes their frequency count. Based on the frequency count of the keywords of positive and negative, automatically it will apply the reviews on test data and classifies the positive and negative reviews from the customers.

1. **DATASET:**

UBER sentiment analysis requires some data or reviews for the customers which contains both positive and negative reviews. For the data we will develop the FACEBOOK API or TWITTER API where we can pull the reviews of UBER based on the customer feelings about a post. But development of this API is very difficult as they were restricted from pulling the comments of the UBER from FACEBOOK. So we used OUTWIT to pull the comments. Outwit is a software available on web for we scraping. This uses some multiple hash tag commands in order to pull the data. Our team mate Ajay Reddy is more concentrated on pulling the comments. Using the OUTWIT we got around 100+ reviews, we filtered junk from those reviews and finally we got twenty useful reviews which shows exact feelings from customers on UBER. Some example reviews that we scraped using OUTWIT are:

* Gud app Recent update made this app bad. Always no cars available n surging in uberx option.. **POSITIVE**
* Very poor service- I have coupons but after completing my ride Uber has charged cash, executives told that my coupon deemed invalid. Poor service, bad drivers. Total disgusting. **NEGATIVE**
* Good service. Good app-I like the app and love the service. The app should really add a tipping function though **POSITIVE**
* Love Uber- Love uber every driver I've rode with is so professional and fun to chat with, I never have a problem keep up the good work! **POSITIVE**

1. **Research Questions:**

UBER sentiment analysis mainly works on predicting the reviews of customers based on the feelings. The research questions are as follows:

* How to predict the feelings or experiences of customers using training data and clustering?
* How to avoid the words that is most used, that is stop words?
* What are the overall features of the customers on UBER service?
* How this prediction process runs on different keywords?
* How this training works based on the data we give as input?
* What will be the different predictions of positive and negative keywords based on the code?

1. **Models and Analysis:**

In this development process we tried to use algorithms and a way different analysis of our Uber data but we got 50% of the prediction with those algorithms. In order to achieve a success rate of the prediction we used some java code with TFIDF calculations and found 90% of the prediction values. In this we used some specific keywords to give a TFIDF frequency values based on that it will predict the reviews to be Positive or Negative.

1. **Code and Application:**

We use java as our main source to develop the clustering code for UBER analysis. We started researching about the algorithms which involve in clustering process. But as a default algorithm we use KMEANS algorithm in order to perform this analysis. Our code was unsuccessful when we used algorithms so we wrote code based on TFIDF score calculations, based on the JAVA code we wrote the reviews will give the positive or negative reviews.

1. **Project Management:**

We are three people who are working on this project as AJAY GADDAM (0513), KIRAN KUMAR (0458), and ABHINAND SHARMA YAMAVARAM (0592). In this AJAY GADDAM as we were restricted in pulling comments using FACEBOOK API, we planned to use OUTWIT which is web available software to scrape the comments on UBER from FACEBOOK and TWITTER. ABHINAND SHARMA is concentrating more on the code development for the prediction process. KIRAN KUMAR will be working on testing the pulled comments and final resting and reporting of the entire documentation. This way we participated in each and every part of the project development process. Since we had 1000+ comments it took more time to remove the junk from the comments.

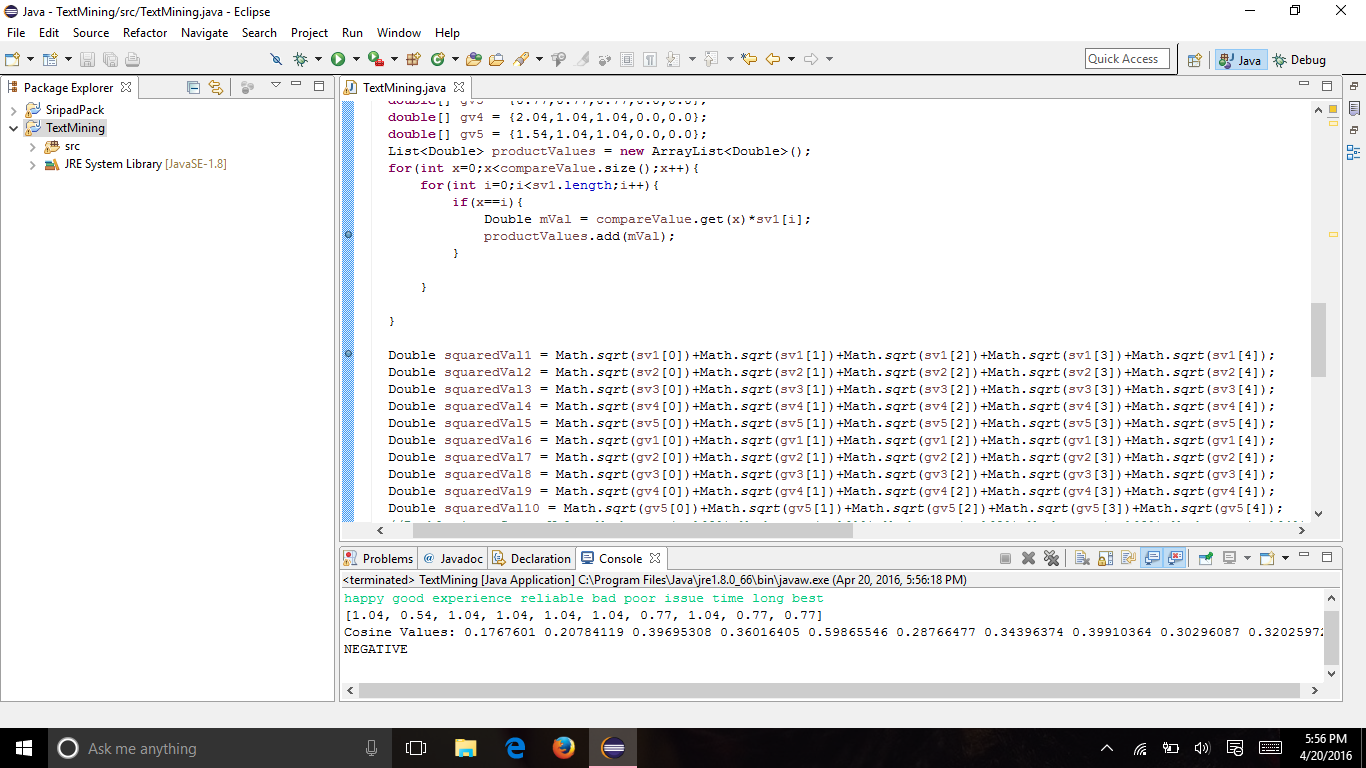
1. **Team Management:**

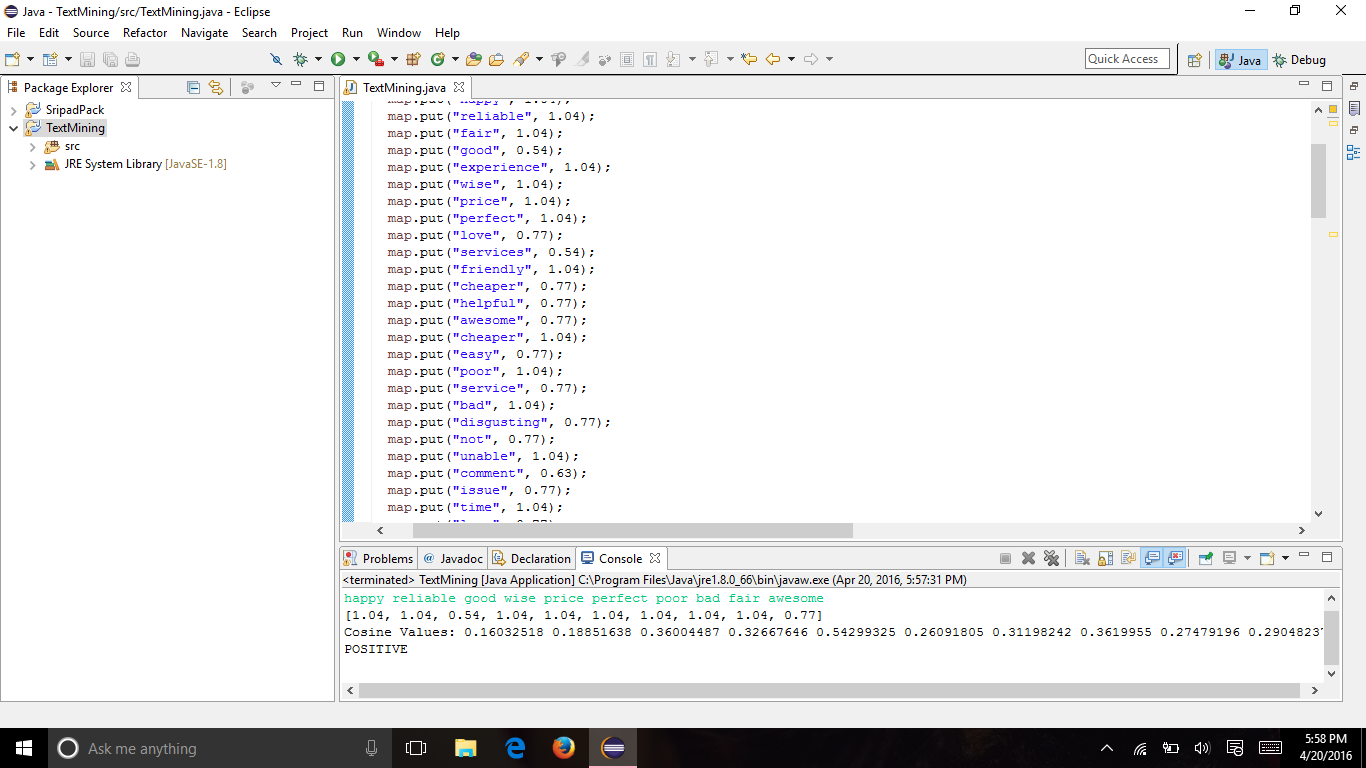
|  |  |  |
| --- | --- | --- |
| **TEAM MEMBER** | **ROLES AND SKILLS** | **CONTRIBUTION** |
| ABHINAND SHARMA(0592) | Coding for sentiment classification | Development of entire sentimental analysis code |
| AJAY GADDAM(0513) | Usage of OUTWIT | Outwit usage in order to scrape the reviews |
| KIRAN KUMAR ADEPU(0458) | Testing and Reporting Data testing | Final testing; Reviewing and Reporting |

1. **Check Points And Deliverables:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Checkpoint date** | **Expected Deliverable** | **Responsible team member(s)** | **Checkpoint results** |
| 03/02/2016 | OUTWIT data scraped on UBER from FACEBOOK and TWITTER | ABHINAND, AJAY, KIRAN | We used a web scraper called OUTWIT in order to scrape the UBER reviews from twitter. All our team members did a research on Web scraper, and twitter reviews |
| 03/22/2016 | Classifying datasets | ABHINAND, KIRAN | Classifying the keywords which represent positive and negative expression in the reviews, We pulled 100+ reviews in which more junk reviews were removed. Abhinand worked on filtering the useful comments, Kiran worked on classifying the positive and negative comments |
| 04/10/2016 | Development of classifiers code | ABHINAND, KIRAN, AJAY | We tried to develop the required code using JAVA, but we faced problems in developing the algorithms, but even though we are in a process of developing a code based on NLTP frequency ranges of each words which will give the positive or negative results based on the frequency values. We three are working on this in order to develop as soon as possible. |
| 04/25/2016 | Reporting | ABHINAND, KIRAN | Final testing and reporting of final data and results |

1. **Final Output:**





1. **Code:**

TextMining.java

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

package datamining;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Scanner;

import java.io.\*;

/\*\*

\*

\* @author ABHINANDSHARMA

\*/

public class Textmining {

/\*\*

\* @param args the command line arguments

\*/

public static void main(String[] args) {

// TODO code application logic here

Map<String,Double> map = new HashMap<String,Double>();

map.put("happy", 1.04);

map.put("reliable", 1.04);

map.put("fair", 1.04);

map.put("good", 0.54);

map.put("experience", 1.04);

map.put("wise", 1.04);

map.put("price", 1.04);

map.put("perfect", 1.04);

map.put("love", 0.77);

map.put("services", 0.54);

map.put("friendly", 1.04);

map.put("cheaper", 0.77);

map.put("helpful", 0.77);

map.put("awesome", 0.77);

map.put("cheaper", 1.04);

map.put("easy", 0.77);

map.put("poor", 1.04);

map.put("service", 0.77);

map.put("bad", 1.04);

map.put("disgusting", 0.77);

map.put("not", 0.77);

map.put("unable", 1.04);

map.put("comment", 0.63);

map.put("issue", 0.77);

map.put("time", 1.04);

map.put("long", 0.77);

map.put("best", 0.63);

map.put("bad", 1.04);

Scanner scanner = new Scanner(System.in);

System.out.println("Please Enter the terms ");

String value=scanner.nextLine();

String[] splitValue = value.split(" ");

List<String> splitList1 = new ArrayList<String>(Arrays.asList(splitValue));

List<Double> compareValue = new ArrayList<Double>();

for(String val:splitList1){

for(Map.Entry<String, Double> mapValues:map.entrySet()){

if(mapValues.getKey().equalsIgnoreCase(val)){

compareValue.add(mapValues.getValue());

}

}

}

System.out.println(compareValue);

double[] sv1 = {1.04,1.04,2.08,2.08,1.04};

double[] sv2 = {1.04,0.77,1.08,1.54,0.77};

double[] sv3 = {1.07,0.77,0.54,0.0,0.0};

double[] sv4 = {1.04,1.04,0.77,0.0,0.0};

double[] sv5 = {0.77,0.77,0.0,0.0,0.0};

double[] gv1 = {0.77,0.77,1.04,0.77,0.0};

double[] gv2 = {1.54,1.04,0.63,0.0,0.0};

double[] gv3 = {0.77,0.77,0.77,0.0,0.0};

double[] gv4 = {2.04,1.04,1.04,0.0,0.0};

double[] gv5 = {1.54,1.04,1.04,0.0,0.0};

List<Double> productValues = new ArrayList<Double>();

for(int x=0;x<compareValue.size();x++){

for(int i=0;i<sv1.length;i++){

if(x==i){

Double mVal = compareValue.get(x)\*sv1[i];

productValues.add(mVal);

}

}

}

Double squaredVal1 = Math.sqrt(sv1[0])+Math.sqrt(sv1[1])+Math.sqrt(sv1[2])+Math.sqrt(sv1[3])+Math.sqrt(sv1[4]);

Double squaredVal2 = Math.sqrt(sv2[0])+Math.sqrt(sv2[1])+Math.sqrt(sv2[2])+Math.sqrt(sv2[3])+Math.sqrt(sv2[4]);

Double squaredVal3 = Math.sqrt(sv3[0])+Math.sqrt(sv3[1])+Math.sqrt(sv3[2])+Math.sqrt(sv3[3])+Math.sqrt(sv3[4]);

Double squaredVal4 = Math.sqrt(sv4[0])+Math.sqrt(sv4[1])+Math.sqrt(sv4[2])+Math.sqrt(sv4[3])+Math.sqrt(sv4[4]);

Double squaredVal5 = Math.sqrt(sv5[0])+Math.sqrt(sv5[1])+Math.sqrt(sv5[2])+Math.sqrt(sv5[3])+Math.sqrt(sv5[4]);

Double squaredVal6 = Math.sqrt(gv1[0])+Math.sqrt(gv1[1])+Math.sqrt(gv1[2])+Math.sqrt(gv1[3])+Math.sqrt(gv1[4]);

Double squaredVal7 = Math.sqrt(gv2[0])+Math.sqrt(gv2[1])+Math.sqrt(gv2[2])+Math.sqrt(gv2[3])+Math.sqrt(gv2[4]);

Double squaredVal8 = Math.sqrt(gv3[0])+Math.sqrt(gv3[1])+Math.sqrt(gv3[2])+Math.sqrt(gv3[3])+Math.sqrt(gv3[4]);

Double squaredVal9 = Math.sqrt(gv4[0])+Math.sqrt(gv4[1])+Math.sqrt(gv4[2])+Math.sqrt(gv4[3])+Math.sqrt(gv4[4]);

Double squaredVal10 = Math.sqrt(gv5[0])+Math.sqrt(gv5[1])+Math.sqrt(gv5[2])+Math.sqrt(gv5[3])+Math.sqrt(gv5[4]);

//Double inputSquareVal = Math.sqrt(sv1[0])+Math.sqrt(sv1[1])+Math.sqrt(sv1[2])+Math.sqrt(sv1[3])+Math.sqrt(sv1[4])+Math.sqrt(sv1[5])+Math.sqrt(sv1[6]);

//Map<String, Double> compareValue;

Double cSquR = Math.sqrt(compareValue.get(0))+Math.sqrt(compareValue.get(1))+Math.sqrt(compareValue.get(2))+Math.sqrt(compareValue.get(3))+Math.sqrt(compareValue.get(4))+Math.sqrt(compareValue.get(5))+Math.sqrt(compareValue.get(6));

Double iMul = productValues.get(0)+productValues.get(1)+productValues.get(2)+productValues.get(3)+productValues.get(4);

Float cosine1 = (float) (iMul/(squaredVal1\*cSquR));

Float cosine2 = (float) (iMul/(squaredVal2\*cSquR));

Float cosine3 = (float) (iMul/(squaredVal3\*cSquR));

Float cosine4 = (float) (iMul/(squaredVal4\*cSquR));

Float cosine5 = (float) (iMul/(squaredVal5\*cSquR));

Float cosine6 = (float) (iMul/(squaredVal6\*cSquR));

Float cosine7 = (float) (iMul/(squaredVal7\*cSquR));

Float cosine8 = (float) (iMul/(squaredVal8\*cSquR));

Float cosine9 = (float) (iMul/(squaredVal9\*cSquR));

Float cosine10 = (float) (iMul/(squaredVal10\*cSquR));

System.out.println("Cosine Values: "+cosine1+" "+cosine2+" "+cosine3+" "+cosine4+" "+cosine5+" "+cosine6+" "+cosine7+" "+cosine8+" "+cosine9+" "+cosine10);

List<Float> list = new ArrayList<Float>();

list.add(cosine1);

list.add(cosine2);

list.add(cosine3);

list.add(cosine4);

list.add(cosine5);

list.add(cosine6);

list.add(cosine7);

list.add(cosine8);

list.add(cosine9);

list.add(cosine10);

int x = 0;

int y = 0;

for(Float val : list){

if(val<=0.3){

x = x+1;

}else{

y=y+1;

}

}

if(x<y){

System.out.println("NEGATIVE");

}else{

System.out.println("POSITIVE");

}

}

}

1. **References:**

Combining Knowledge and Data Mining to Understand Sentiment - A Practical Assessment of Approaches

<http://www.sas.com/en_us/offers/sem/combining-knowledge-data-mining-to-understand-sentiment-105008.html?keyword=sentiment+analysis&matchtype=e&publisher=google&gclid=CjwKEAjw0pa5BRCLmoKIx_HTh1wSJABk5F_4XhiJyhNbHyrZtZG7XfeZJDyndxPBIw03zD3iyXejShoCUEDw_wcB>

Social Media Sentiment Analysis

<http://go.alchemyapi.com/sentiment-analysis?gclid=CjwKEAjw0pa5BRCLmoKIx_HTh1wSJABk5F_4qiHq0Nc3tDOp_bh6bstDQhivG7ORjbCuR_XtCy8ftRoCgXbw_wcB#utm_source=google&utm_medium=cpc&utm_term=sentiment%20analysis&utm_content=social&utm_campaign=sentiment-analysis>

Sentiment Analysis

<http://nlp.stanford.edu/sentiment/>

Sentiment Analysis API

<http://www.alchemyapi.com/products/alchemylanguage/sentiment-analysis>

Sentiment analysis using product review data

<http://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2>

Sentiment Analysis

<https://www.lexalytics.com/technology/sentiment>