Technical Paper
Group 1

1. NLTK-Weight calculation

1.1. calculation method

When search text file, normally we use vector space model, in this model, a word x is a vector in predefined word set and terminology set, we call it □(x); when we normalize all the vector, we can ignore the length of each text information. The distance of two vector can be calculated by the interior product of this two vectors.

1.2. Implementation

If we want to calculate the weight of each text data, we must normalize each word, so all the data should be extract the root according his position in sentence. The method of how to extract root has been introduced in previous report.

After no normalize each tweeter, and eliminate useless character, the sample text data should be like this:

```
653928219951955098 1 just play volleyball and basketball in the evening so much my foot hurt badly n i m exhaust but i want to finish
653928219951955867 1 hit the gym twice today and also play basketball in the evening i hope i get into this routine
653928219951955675 1 i ve play way too much basketball this evening something tell me i m go to regret this in the morning ballnight
```

As we can see, all the word has been processed, the useless character has been eliminated like '#,%()' and all the word has been transformed into its root. After that, we calculate the weight according to method introduced in 'An Introduction to Support Vector Machine and Other Kernel based Learning ' Nello Cristianini and John Shawe-Taylor.

The source code of the calculation in .py file is in this part:

```
weight=[]  # the list which will be sotred into the database
tmp=[]
for i in range(0,feature_num):
    for j in range(0,row_data):
        if tmp2[i][j]==0:
            tmp.append(0)
        else:
            tmp.append(tmp2[i][j]*math.log(total_List[i]))
    weight.append(tmp)
    tmp=[]
```

'tmp2' array stores the times single feature appear in a single tweeter, and 'total_List' array stores times this feature appears in whole data set. I discard the constant in the formula because SVM can take the value bigger than one.

After calculation, the sample output should be like this:

```
1 1  0  0  0  0  11.2908937953  0  0  0  0  0  11.4601995659  0  0  0  0
2 1  0  0  0  0  11.2908937953  0  0  0  0  0  5.73009978297  0  0  0  0
3 1  0  0  0  0  11.2908937953  0  0  0  0  0  5.73009978297  5.17614973257  0  0  0
4 1  0  0  0  0  5.64544689764  0  0  0  0  0  5.73009978297  5.17614973257  0  0  0
```

The first column, is the result of each tweeter, 0 means 'watcher group' and 1 means 'self-involvement' group. After that is 15 feature, the detailed information about feature selection, please see the feature part.

2. SVM Project

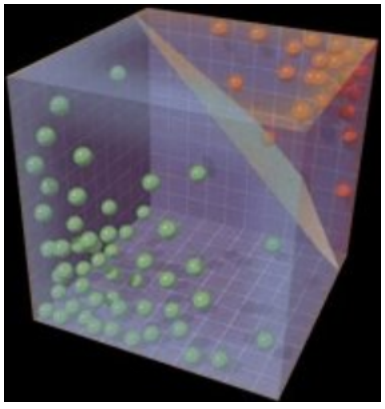
1. Model creation

In SVM project, the first step is to create a model, we use this model to predict testing data in future, so this model is very critical. This this part code, we set the important parameter in SVM, these parameters determine the type of SVM[1].

[1] <A Practical Guide to Support Vector Classification> Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin,Department of Computer Science, National Taiwan University, Taipei 106, Taiwan

```
param.svm_type = C_SVC;
param.kernel_type = RBF;
param.degree = 3;
param.gamma = 0.0001;
param.coef0 = 0;
param.nu = 0.5;
param.cache_size = 100;
param.C = 10;
param.eps = 1e-5;
param.shrinking = 1;
param.probability = 0;
param.nr_weight = 0;
param.weight_label = NULL;
param.weight = NULL;
```

With all these parameter, we create a '0-1' type support machine, like the figure below, every point in the space is the tweeter data. We use the feature weight and core function in the SVM to make them into 3-dimension vector[2]. Using the result, the SVM can calculate a most suitable dimension which separate all the vector in the space. And it can calculate the distance for each vector to the dimension.



After the dimension has been decided, when comes the vectors, SVM will calculate each weight and see how far to the dimension and which side it belongs to. More detailed information, please refer to document about SVM and there are many good brief introduction website[3].

2. Prediction

   After the model has been created, we can use this model to predict the testing data. The testing data must in the fixed format so that the software can get the data from it. The data should be in following format:

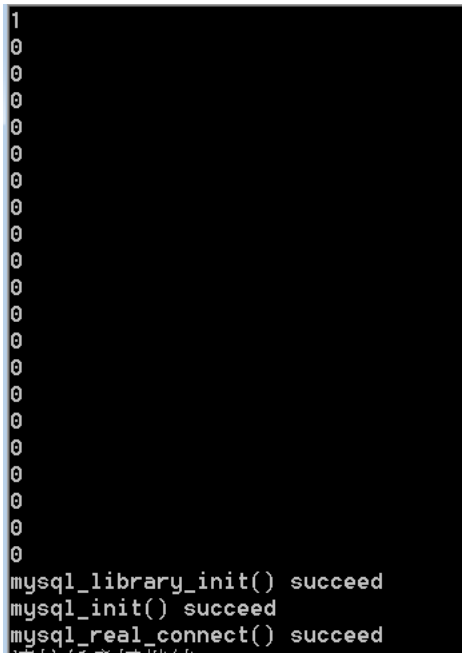
```
653928219951955124  0  0  0  0  5.76519110278  0  0  0  0  0  11.6042367508  0  0  0  0
653928219951955976  0  0  0  0  0  0  0  0  0  4.48863636973  5.80211837538  0  0  0  0
653928219951955233  0  0  0  0  0  0  0  0  0  0  5.80211837538  0  0  0  0
653928219951955976  0  0  0  0  5.76519110278  0  0  0  0  0  5.80211837538  0  0  0  0
```

   The first column is the tweeter id, the prediction project will discard it automatically. After that is 15 features, which comes from the weight calculation (you can find the detailed in previous section). The prediction project will give you the result according to your parameter and store the result into database.

---

[2] 'An Introduction to Support Vector Machine and Other Kernel-based Learning ' Nello Cristianini and John Shawe-Taylor

[3] http://blog.sina.com.cn/s/blog_67ad2e480100zaue.html



After showing the result, they will be stored into database automatically. About the database information, please see the next section.

3. Database Update

Since our display part need different kind result to show the ratio and proportion for extent of participation of certain sport. The database must be the intermediate part of the project, we create other different kinds of table to store the result, here is the code for basketball.

```
CREATE TABLE IF NOT EXISTS `basketball_ratio` (
  `id` int(9) NOT NULL,
  `tweet_id` bigint(20) unsigned NOT NULL,
  `inType` int(1) NOT NULL,
  `inserttime` varchar(40) NOT NULL,
  `otherInfo` varchar(30) DEFAULT NULL
) ENGINE=MyISAM AUTO_INCREMENT=258 DEFAULT CHARSET=utf8;
```

'id' will increase automatically, the 'tweet id' can relate to original tweeter information. 'inType' is the result value and other are self-explained.

After the SVM calculation, all the result will be stored into the table. Here is the example for basketball:

	id	tweet_id	inType	inserttime	otherInfo
编辑 复制 删除	213	653928219951955124	1	Fri Dec 04 16:12:45 2015	NULL
编辑 复制 删除	214	653928219951955976	1	Fri Dec 04 16:12:45 2015	NULL
编辑 复制 删除	215	653928219951955233	1	Fri Dec 04 16:12:45 2015	NULL
编辑 复制 删除	216	653928219951955976	1	Fri Dec 04 16:12:45 2015	NULL
编辑 复制 删除	217	653928219951955233	1	Fri Dec 04 16:12:45 2015	NULL
编辑 复制 删除	218	653928219951955533	1	Fri Dec 04 16:12:45 2015	NULL
编辑 复制 删除	219	653928219951955976	1	Fri Dec 04 16:12:45 2015	NULL
编辑 复制 删除	220	653928219951955543	1	Fri Dec 04 16:12:45 2015	NULL
编辑 复制 删除	221	653928219951955765	1	Fri Dec 04 16:12:45 2015	NULL
编辑 复制 删除	222	653928219951955122	0	Fri Dec 04 16:12:45 2015	NULL
编辑 复制 删除	223	653928219951955233	1	Fri Dec 04 16:12:45 2015	NULL
编辑 复制 删除	224	653928219951955923	1	Fri Dec 04 16:12:45 2015	NULL

We have different table for different sports. If we want to calculate the proportion of self-involvement for basketball, we just add up all the result of 1 and divided by all the record. Use this proportion, we can related to healthy information. We estimate that, the self-involvement ratio should be proportional to healthy condition.

4. Feature Selection

The goal of our system is to learn to understand what people expressed like a AI machine. So we experience the process of feature selection in our main work and over through many issues and improvement during this period.

In the beginning of this process, we adjust our different features into each feature set which means that each set is consisted of the same type keywords or symbols. So we build our feature sets for SVM classification. As more and more twitter we collected, we gradually found out that there is a truth in each twitter. No more than 140 characters in each tweet. Tweet is a short but complex text in the ways of expressing users' feeling or situation. Tweets are kind of unstructured text and consisting of lots of internet slang or Acronyms in Hashtag. Also, Hashtag is a character feature in the each tweets and same as in our data filtering process.

Under this improvement, we kept going and found that there are usually 5 characters each word in each tweet in statistic. Then 25~30 words in a tweet(normally a long tweet) in average. So, in the end, there are only 10~15 "real" words in each tweet. "Real" here means that represent the real feeling and content in this tweet. Therefore, we built our 15 feature sets in our analysis system.

Second phase in this section is to set and fix each feature set by increasing or decreasing components. The table below is an example of our feature set that listing our components in the first-fifth feature set. Because each component could be positive, negative or ambiguous to self-involvement(player) type or others(audience) type in its own set. Such as verb words in set #1, "play" is a action and "watch" is also a action and they both could effect our results by existing in the same tweet. So here we consider many types of words or terminologies which could be related to our specific sport and then contain them in our whole system. While calculating the result of quantitive data, it's not simple like the method I said. We also utilize the method of calculating the weight of each feature and combine them into final

result.

In the next phase, we could consider that move to contain more bigram words or feature into

Category	feature1	feat 2	feat 3	feat 4	feat 5	feat 6	feature7-15
“player”	verb			weather	mood	terminology	…
“audience”	verb	player name	venue		mood	terminology	…

our system. Also more internet slangs and acronyms would be analyzed and selected carefully and then make them useful to training model. We still study a method named Latent Semantic Analysis and we think it would be suitable to execute its algorithm in the process of analyzing Hashtags.

5. We use the TextBlob to implement the emotional analysis. TextBlob is a Python (2 and 3) library for processing textual data. It provides a consistent API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, and more. TextBlob stands on the giant shoulders of NLTK and pattern, and plays nicely with both.

In our project, we use TextBlob to do the emotional analysis on every exercise and sports related tweets and classify them into either a positive tweet or a negative one. By doing so, we are able to see how many of people talking specific sports are having negative feelings.

The code below is a simple Python example program using TextBlob API to get emotional analysis on tweets.

Android application

1. setting up server

Here are some parts of the codes to set up a database, you can see that I’m setting the port number to 9999 so that my android application will be able to connect to this data base. I’m using JDBC, kind of like a java API to set up this server. That’s why you can see that I’m calling function name JDBC. And that I’m setting up JDBC driver name and database URL so that my server is also connected to the database, which is implemented by mysql.

```
public class Server {
     private static final int PORT = 9999;
    private static List<Socket> mList = new ArrayList<Socket>();
    private static ExecutorService mExecutorService = null; //thread pool
    private static JSONArray userarray = new JSONArray();
```

```
// JDBC driver name and database URL
static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
static final String DB_URL = "jdbc:mysql://localhost/";

//  Database credentials
static final String USER = "root";
static final String PASS = "";
static final String DBNAME = "SEHEALTH";
```

Here, I just show you the overall structure of this application, the specific codes are contained in the folder called codes.

2. Connecting database to the server

Here, I am connecting the database to the server. If there is no existing database, I'll just create a new one.

```
//STEP 3: Open a connection
System.out.println("Connecting to database...");
conn = DriverManager.getConnection(DB_URL, USER, PASS);

//STEP 4: Execute a query
System.out.println("Creating database...");
stmt = conn.createStatement();

//create the database if not exists
String sql = "CREATE DATABASE IF NOT EXISTS "+DBNAME;
stmt.executeUpdate(sql);

        //create the tables if not exists
```

3. Android application

This is the screen shot of the Android application folder.

名称	修改日期	类型	大小
.gradle	11/27/2015 1:10 ...	文件夹	
.idea	11/27/2015 1:11 ...	文件夹	
app	11/27/2015 1:11 ...	文件夹	
build	11/27/2015 1:10 ...	文件夹	
gradle	11/27/2015 1:10 ...	文件夹	
.gitignore	11/27/2015 1:10 ...	GITIGNORE 文件	1 KB
build.gradle	11/27/2015 1:10 ...	GRADLE 文件	1 KB
gradle.properties	11/27/2015 1:10 ...	PROPERTIES 文件	1 KB
gradlew	11/27/2015 1:10 ...	文件	5 KB
gradlew.bat	11/27/2015 1:10 ...	Windows 批处理文件	3 KB
local.properties	11/27/2015 1:10 ...	PROPERTIES 文件	1 KB
SEHealth.iml	11/27/2015 1:11 ...	IML 文件	1 KB
settings.gradle	11/27/2015 1:10 ...	GRADLE 文件	1 KB

Here is part of the Android application codes.

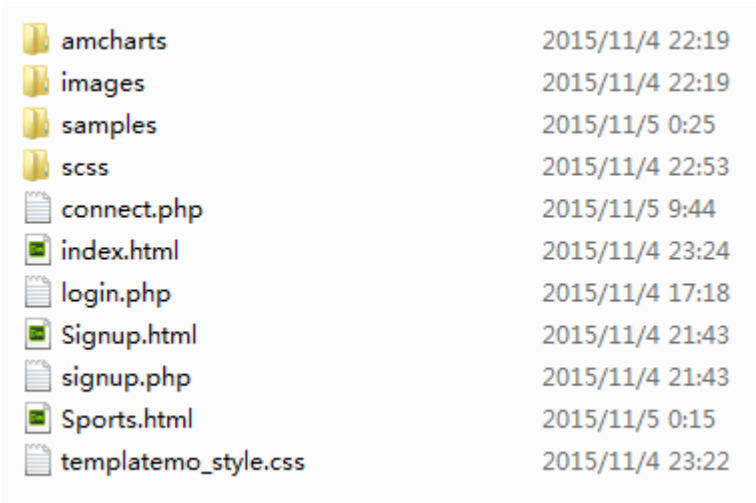
```
?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.zzj.sehealth" >
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
       <activity
            android:name=".activity.ConnectActivity"
            android:label="@string/app_name" >
```

```
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".activity.Login"
                  android:label="LoginActivity"
                  android:launchMode="singleTask">
        </activity>
        <activity android:name=".activity.SignUp"
                  android:label="SignUpActivity"
                  android:launchMode="singleTask">
        </activity>
        <activity android:name=".activity.MainActivity"
                  android:label="MainActivity"
                  android:launchMode="singleTask">
        </activity>
        <service android:name=".service.SocketService"
                 android:enabled="true"
                 android:exported="true"/>
    </application>

</manifest>
```

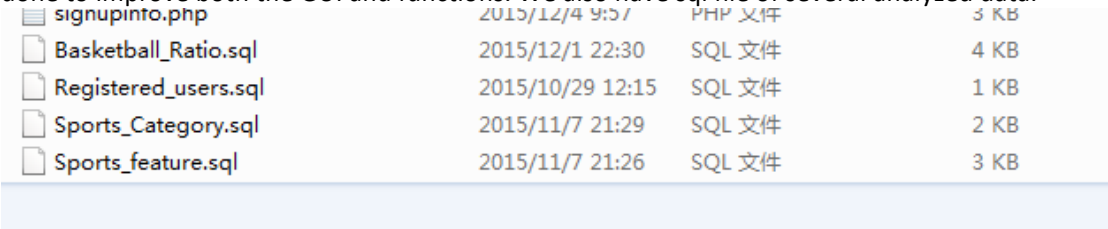
Website:

1. Linking to data

This is the screen shot of website, including all the current html pages, css tables, JavaScript embedded charts and php pages.



The whole structure of website is as shown, though it still needs time to improve, we just offer the web view and simple function of website charts based on current data. Further work will be done to improve both the GUI and functions. We also have sql file of several analyzed data.



These files are what cone from SVM and nltk, which will be used to display charts in website.

2. Getting data

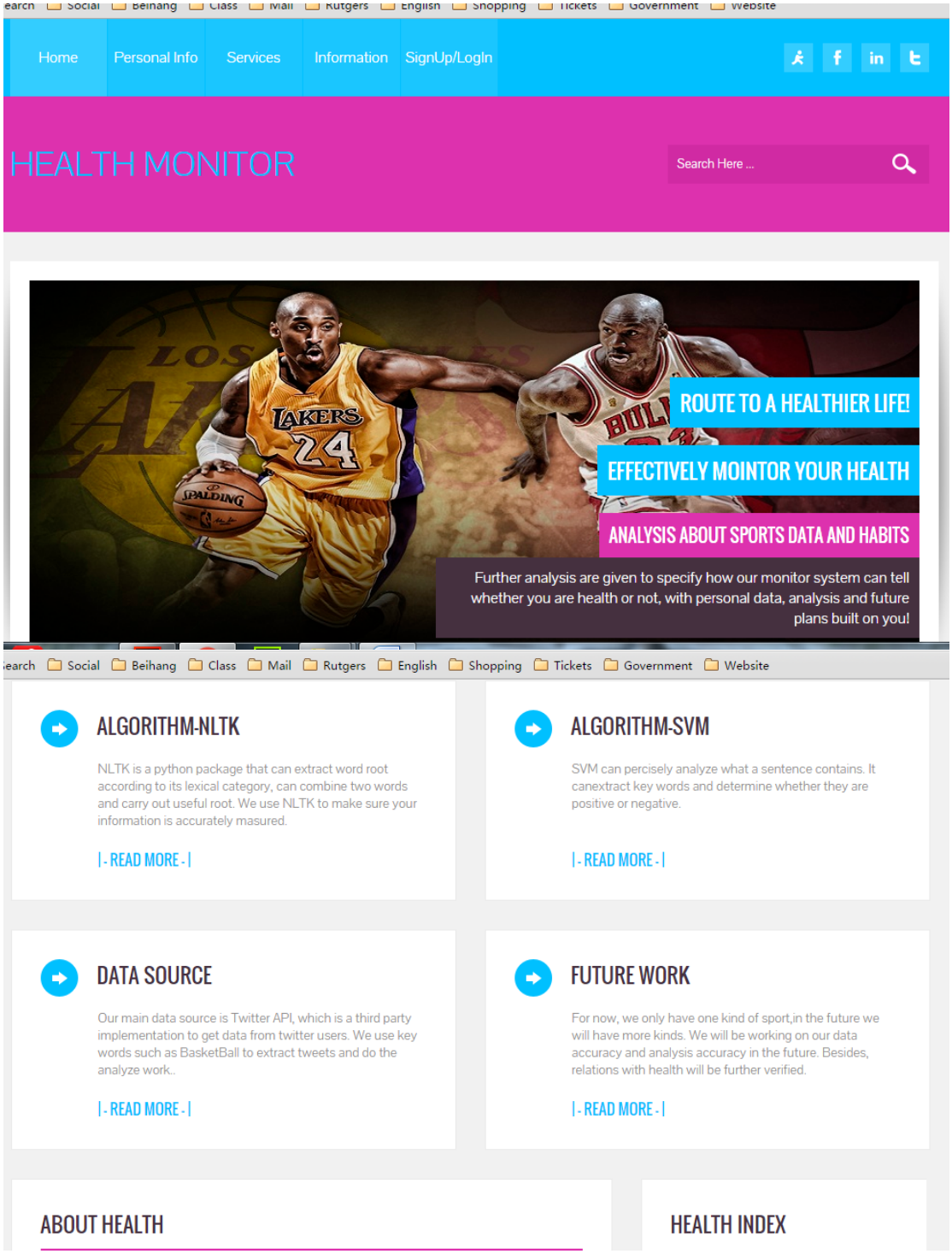
Receiving data follows a sequence as:

1 Data grabbed and stored in our database;
2 Using algorithms to analyze the data collected;
3 Website offering data collected and analyzed;
Since we're using php language to fetch data from database, so all the fetching are through php to Mysql.
Some of the php source codes are as follows:

```
<?php
// Connect to MySQL
$link = mysql_connect( 'localhost', 'root', 'a' );
if ( !$link ) {
die( 'Could not connect: ' . mysql_error() );
}
// Select the data base
$db = mysql_select_db( 'test', $link );
if ( !$db ) {
die ( 'Error selecting database \'test\' : ' . mysql_error() );
}
// Fetch the data
$query = "
SELECT *
FROM my_chart_data
ORDER BY category ASC";
$result = mysql_query( $query );
// All good?
if ( !$result ) {
// Nope
$message = 'Invalid query: ' . mysql_error() . "\n";
$message .= 'Whole query: ' . $query;
die( $message );
}
// Print out rows
while ( $row = mysql_fetch_assoc( $result ) ) {
echo $row[''] . ' | ' . $row['value1'] . ' | ' .$row['value2'] . "\n";
}
// Close the connection
mysql_close($link);
?>
```

This is just an example php for getting data out of sql through Mysql, embedding php has made it quite easy to read data from database.

3.Information Display
3.1 Showing data and algoruthm



Our system includes several parts, a homepage named index indicating how the website is constructed. Index page includes website map and direction, it contains three subpages.

The three subpages are personal information, information, service and signup/login. Three kinds of graph are given using the data we have now, which are not accurate because of the small size of the sample. In further phase, we will be focusing on data accuracy. This is the graph based on our current data.

3.2 Linking to database

The data extracted from twitter using API is stored in a database, soon executed by the algorithm of NTLK and KNN, extracting the meaning from those sentences from twitter. After analyzing the

meaning the sentence, the results are put in the database as a table or linked list in the same database of mysql, it is called hma. Using php, we can extracting data to website and make it accessibly visible to both users and administrators as ourselves.
The part of code that using data from database to show them on websites using multiple dynamic effects are given below:

```
var chartData = [
{
"Region Zip Code": 'BasketBall',
"People Involved": 23.5,
"Eeople's Intension": 18.1
},
{
"Region Zip Code": 'FootBall',
"People Involved": 26.2,
"Eeople's Intension": 22.8
},
{
"Region Zip Code": 'Running',
"People Involved": 30.1,
"Eeople's Intension": 23.9
},
{
"Region Zip Code": 'Swimming',
"People Involved": 29.5,
"Eeople's Intension": 25.1
},
{
"Region Zip Code": 'Tennis',
"People Involved": 24.6,
"Eeople's Intension": 25
}
];
```

From this example, we can see that we can define the data source using var and change the variable name if we want.