CHAPTER 3
FRAMEWORK FOR EVALUATING MEDICAL BLOG AND CAMERA OPINIONS

3.1 INTRODUCTION

Opinion Mining (OM) is a kind of natural language processing for the purpose of recording attitudes and sentiments of the common people regarding certain topics, products or service. OM recognizes subjectivities as well as objectivities of texts and classifies them with regard to the opinions orientation of subjective texts (Mishra & Jha 2012).

Cameras are popular in social as well as computing landscapes and implanted in customer gadgets such as smart phones, tablets, laptops as well as wearable gadgets like Google Glass, Narrative Clips and Aerographers. They are on the fringe of becoming a ubiquitous device. Opinion holders are persons or enterprises holding a particular opinion. In product review sites, forums or blog posts, opinion holders are the writers of those posts.

Online reviews express opinions about a product or service and users evaluate a product or service based on these opinions before buying or using the product. Due to the huge amount of reviews available in different websites, it is hard to comprehend all the opinions. Opinion mining summarizes and the polarity of the various reviews which helps in gaining a overall picture about a product or service. The Sentiment is classified as negative, neutral or positive on retrieving the information from the review. Various techniques such as clustering, supervised learning methods classify
sentiment polarity. Sentiment classification has been widely researched and several approaches are surveyed in literature (Cambria 2013). The efficacy of the feature extraction methods and classification algorithms for classifying cameras reviews were investigated. Opinions expressed on cameras are taken from Amazon website. TDF×IDF is utilized for the extraction of features from camera reviews. Features transformation is undertaken by using PCA and kernel PCA. Naïve Bayes, K-Nearest Neighbour classifiers and CART algorithms performance evaluations are investigated.

3.2 METHODOLOGY

The flowchart of the methodology followed is shown in Figure 3.1.
3.2.1 Data set

Blogs: With internet being utilized increasingly, blogs and blog sites are also becoming more popular. Blogs are now the most widely used method for expressing own opinions and views. A blogger records the generic events in day to day life and mention their opinions and sentiments regarding them in blogs. Several blogs comprise reviews of various products, policies, news topics and so on. Blogs are typically utilized as sources of opinions in several researches linked to sentiment analyses (Tang et al. 2009).

Camera Dataset: Opinions are collected from amazon. Two hundred and twenty five each of positive and negative reviews are used. Some examples of the positive and negative reviews are presented here.

Positive Reviews: “This is second Sony cyber shot digital camera, although I have purchased Kodak Easy Shares for family members. I loved the first one (only 3.2 MGP). This camera is perfect for the not-so-tech-wise consumer. It takes great pictures and has a high quality Zeiss lens. Most of all it is easy to use, especially for the beginner and intermediate user. It stores easily in a pocket and I love the color choices Sony gives you! The review pictures button is a little small, but get used to it quickly.”

Huang: little camera - especially for point and shoots users who just want a camera to take snapshots and is not interested in becoming a rocket scientist in order to learn how to operate the camera. A 7.2 MP and fast shutter with reasonable flash for its size, hard to mess up pictures.”

Negative Reviews: “Battery life is terrible if use image stabilizer, expect 25-30 shots on a full battery. Also the camera lacks of an optical view finder very difficult to shoot in sunlight with LCD. Many shots are somewhat bleached out while using auto white balance. Owned a stylus 400 digital prior to this and it is a disappointment rather than upgrade. The only upsides are the
5x optical zoom which is a little choppy and the image stabilizer that kills the battery if left on.”

### 3.2.2 Feature Extraction

- **Stemming**: Stemming is a reference to root word origins. For instance, speak is the root term for speak, speaking and speaks. In many cases, words morphological variants have similar semantic interpretations and are considered equal for IR applications. Due to this reason, many stemming Algorithms, or stemmers, were built to decrease a word to its stem or root form. Hence a query or document’s key terms are denoted by stems and not by original terms. This denotes that a term’s differing variants can be merged to one representative format in addition to reducing dictionary size i.e. number of distinct terms required to represent a document set.

- **Stop word**: A typical stop word list for words without purpose for retrieval, but frequently utilized to compose documents, are built for two main reasons: First, it is possible that a query and document match have their basis in good indexing terms. So, retrieving document which has words like "be", "the" and "your" in related request is not intelligent strategy. These non-significant words denote noise and damage retrieval performance failing to discriminate between relevant and non-relevant documents. Second, it is expected to reduce inverted file size to a range between 30 and 50%.
3.2.3 Feature Transformation

The occurrences of every word in a document are represented through Term Frequency (TF) that is a document specific measure of term importance. A documents collection being considered is a corpus. Many term weighting techniques were proposed in the literature. A document vector represents a vector space model whose components are term weights.

The TF-IDF weight (Term Frequency-Inverse Document Frequency) is used to evaluate how important a word is to a document in a collection or corpus. The importance of TF-IDF increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the TF-IDF weighting is used by search engines as a central tool in scoring and ranking a document's relevance given a user query. IDF is based on counting the number of documents in collection being searched which contain the term in question. If a term occurs in all the documents of the collection, its IDF is zero

$$idf_i = \log \frac{|D|}{|\{j : t_i \in d_j\}|}$$

(3.1)

with $|D|$ : cardinality of D, or the total number of documents in the corpus $|\{j : t_i \in d_j\}|$ : number of documents where the term $t_i$ appears (viz. the document frequency) (that is $n_{i,j} \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore common to use $1 + |\{j : t_i \in d_j\}|$.

TF count is normalized to prevent a bias towards longer documents to give a measure of the importance of the term $t_i$ within the particular document $d_j$. 
\[ tf_{i,j} = \frac{n_{i,j}}{\sum_i n_{i,j}} \]  

(3.2)

Where \( n_{i,j} \) is the number of occurrences of the considered term \( t_i \) in document \( d_j \), and the denominator is the sum of number of occurrences of all terms in document \( d_j \), that is, the size of the document \(|d_j|\).

Alternatively:

\[ \frac{tf_{i,d}}{\max \text{ } tf_d} \]  

(3.3)

where \( \max \text{ } tf_d \) is the max frequency within the document.

In TF-IDF method the term weight is incremented on the basis of frequency in a document and reduced on the basis of frequency across document sets. To assess these impacts, the TF in a document is determined as in equation (3.4):

\[ tf_{i,d} = \frac{\text{number of occurrence of the term } t_i}{\text{number of terms in the document } d} \]  

(3.4)

And the IDF as in equation (3.5):

\[ idf_t = \log \frac{N}{df_t} \]  

(3.5)

By multiplying the TF-IDF weight is determined for a term \( t \) in a document \( d \) as in equation (3.6):

\[ tf - idf_{r,a} = tf_{r,a} \times idf_t \]  

(3.6)
3.2.4 Feature Selection

PCA (Principal Component Analysis) and Kernel PCA (KPCA) is the most commonly used technique for feature selection. PCA (Qu et al. 2013) is a linear dimensionality decreasing method typically used that projects higher dimensional information along principal axes selected by calculating variance. Through the enhancement of principal component analysis, several correlative methods are suggested that possess improved capacity than principal component analysis in particular application domain.

PCA tries to find lower dimensionality linear subspace of original feature space where new features have largest variance. This is called dimensionality reduction, as vector $\bar{x}$ containing initial data and is N-dimensional is lowered to a compressed vector $\overline{c}$ that is M-dimensional, where M<N. A vector $\bar{x}$ is coded into a vector $\overline{c}$ with reduced dimension. Vector $\overline{c}$ is stored, transmitted or processed resulting in vector $\overline{c}^{'}$ capable of being decoded back to a vector $x^{''}$. The last vector is a result approximation which can be reached by storing, transmitting or processing vector $\bar{x}$ . The diagram’s encoder should perform a linear operation, using a matrix $\overline{Q}$ :

$$\overline{c} = \overline{Q}\bar{x}$$  \hspace{1cm} (3.7)

Decoder is also a linear operation, written as a sum of vector elements of $\overline{c}$ multiplied by matrix columns:

$$\bar{x}^{''} = \overline{c}^{T} \overline{Q}^{T} \rightarrow \bar{x} = \sum_{i=1}^{M} c_{i} \overline{q}_{i}$$  \hspace{1cm} (3.8)

Kernel PCA: The fundamental premise of KPCA is to map higher dimensional information to features spaces through non-linear methods and perform PCA in it. KPCA due to its non-linearity, possesses the great quality
of kernel methods which establish links between higher dimensional and lower dimensional information with kernel functions not needed determining particular relationship between them.

Kernel Principal Component Analysis (kernel PCA) as a nonlinear generalization of PCA was suggested with the aim being to map given data points from input space $\mathbb{R}^n$ to higher-dimensional (infinite dimensional) features space $F$:

$$\Phi = \mathbb{R}^n \rightarrow F$$

(3.9)

and perform PCA in F. The space F and also mapping $\Phi$ might be complicated. But using so-called kernel trick, it avoids using $\Phi$ explicitly: PCA in F is formulated so that only F’s inner product is needed which is seen as a nonlinear function called kernel function:

$$\mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$$

$$(x, y) \rightarrow k(x, y)$$

(3.10)

3.2.5 Classifiers

Naïve Bayes: It is the statistical classifier based on Bayes theorem which uses a probabilistic method to estimate given information’s class matching it to the class with highest posterior probability.

$$P(C_i | V) = \frac{P(V | C_i)P(C_i)}{P(V)}$$

(3.11)

Where $V = (v_1, \ldots, v_n)$ is document denoted in n dimensional features vector and $c_1, \ldots, c_m$ denotes m class. But it is costly with regards to
computation, when calculating $P(V \mid C_i)$. To decrease calculations, naïve conditional independence assumption of class is done. Thus:

$$P(V \mid C_i) = \prod_{k=1}^{n} P(x_k \mid C_i)$$  \hspace{1cm} (3.12)

**K-Nearest Neighbour Classification:** K-Nearest Neighbour classifier is based on premises that vector space model is similar for similar documents. Training documents are indexed and each associated with corresponding label. A submitted test document is treated like a query retrieving from training set, documents similar to test document. The test document class label is assigned based on distribution of k nearest neighbours. Class label can be refined by adding weights. Tuning k, obtains higher accuracy. Nearest neighbour method is easy to understand and implement:

$$p(x) \approx \frac{k}{NV}$$  \hspace{1cm} (3.13)

Similarly, probability density function $p(x \mid H_i)$ of observation $x$ conditioned to hypothesis $H_i$ is approximated 24. Let us assume $N_i$ is number of patterns associated to hypothesis:

$$H_i, i = 1, \ldots, C \text{ so that } N_1 + \ldots + N_C = N$$  \hspace{1cm} (3.14)

**Classification and Regression Trees (CART):** CART handles numerical and categorical parameters. CART’s benefits include robustness to outliers. Typically splitting algorithms isolates outliers in individual node/nodes. A CART useful feature is that classification or regression trees structure does not vary regarding independent variables monotone
transformations. Any variable can be replaced with logarithm or square root value and tree structure does not change:

\[ i(t) - p_L i(t_L) - p_R i(t_R) \]  

(3.15)

CART selects split maximizing impurity decrease CART methodology has three parts: Maximum tree construction, Choice of correct tree size and New data classification using constructed tree.

3.3 RESULTS AND DISCUSSION

The opinions are collected from Amazon website and 225 positive and 225 negative features are used in this study. Features are extracted using TDF×IDF and Feature transformation is achieved using PCA and kernel PCA.

Accuracy of Naïve Bayes, KNN and CART approaches to classify reviews is evaluated. Experiments are conducted for: Feature extraction using only TDF×IDF, Feature extraction using TDF×IDF and PCA and Feature extraction using TDF×IDF and kernel PCA.

Accuracy of OM is tested through statistical metrics like recall, precision or F1 measure. **Accuracy**, which may be defined as the percentage of sentiments accurately predicted, is a technique of evaluation. Quality of outcomes is also assessed through comparison of 2 standard performance metrics which are recall as well as precision. **Recall** may be described as the proportion of positive sentiments that are accurately identified (Kim et al., 2014).

\[
\text{Recall} = \frac{\text{Positive instance predicted}}{\text{Total positive instances}}
\]
**Precision** is described as the proportion of correct sentiments predicted to overall quantity of instances predicted.

\[
\text{Precision} = \frac{\text{True positive instances predicted}}{\text{Total instances predicted}}
\]

However, this typically decreases precision of results. Generally, recall as well as precision are inversely related. Ideal learning models have high recall as well as precision. At times, they are fused together as F1, that denotes a harmonic means of recall as well as precision:

\[
F1\text{Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}
\]

**Table 3.1 Classification accuracy, Precision and Recall using Medical Blogs (%)**

<table>
<thead>
<tr>
<th></th>
<th>Classification Accuracy (%)</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE using TDFxIDF – NB</td>
<td>74.22</td>
<td>74.295</td>
<td>74.225</td>
</tr>
<tr>
<td>FE using TDFxIDF – CART</td>
<td>77.33</td>
<td>77.35</td>
<td>77.335</td>
</tr>
<tr>
<td>FE using TDFxIDF – K-NN</td>
<td>73.78</td>
<td>73.785</td>
<td>73.78</td>
</tr>
<tr>
<td>FE using TDFxIDF and PCA- NB</td>
<td>75.56</td>
<td>75.555</td>
<td>75.555</td>
</tr>
<tr>
<td>FE using TDFxIDF and PCA- CART</td>
<td>78.44</td>
<td>78.49</td>
<td>78.44</td>
</tr>
<tr>
<td>FE using TDFxIDF and PCA- K-NN</td>
<td>74.67</td>
<td>74.67</td>
<td>74.665</td>
</tr>
<tr>
<td>FE using TDFxIDF and kernel PCA- NB</td>
<td>76.27</td>
<td>76.275</td>
<td>76.275</td>
</tr>
<tr>
<td>FE using TDFxIDF and kernel PCA- CART</td>
<td>79.11</td>
<td>79.195</td>
<td>79.11</td>
</tr>
<tr>
<td>FE using TDFxIDF and kernel PCA - K-NN</td>
<td>75.78</td>
<td>75.775</td>
<td>75.78</td>
</tr>
</tbody>
</table>
Figure 3.2 Classification Accuracy for Medical Blog

Figure 3.2 to 3.4 shows the classification accuracy, precision and recall respectively for Medical Blogs dataset and Figure 3.5 to 3.7 shows the classification accuracy, precision and recall respectively for camera dataset.

From Table 3.1 and Figure 3.2 it is observed that CART performs classification accuracy better than NB and K-NN. Similarly FE using TDFxIDF and kernel PCA-CART performs better than FE using TDFxIDF-NB and FE using TDFxIDF-K-NN. Results shows that the FE using TDFxIDF and kernel PCA-CART performs accuracy better by 2.28% than FE using TDFxIDF-CART and by 0.85% than FE using TDFxIDF and PCA-CART.

From Table 3.1 and Figure 3.3 it is observed that CART performs precision better than NB and KNN. Similarly FE using TDFxIDF and kernel PCA-CART performs better than FE using TDFxIDF-NB and FE using TDFxIDF-KNN. Results shows that the FE using TDFxIDF and kernel PCA-
CART performs precision better by 2.36% than FE using TDFxIDF-CART and by 0.89% than FE using TDFxIDF and PCA-CART.

**Figure 3.3 Precision for Medical Blog**

![Precision Graph](image1)

**Figure 3.4 Recall for Medical Blog**

From Table 3.1 and Figure 3.4 it is observed that CART performs recall better than NB and KNN. Similarly FE using TDFxIDF and kernel
PCA-CART performs better than FE using TDFxIDF-NB and FE using TDFxIDF-KNN. Results shows that the FE using TDFxIDF and kernel PCA-CART performs recall better by 2.27% than FE using TDFxIDF-CART and by 0.85% than FE using TDFxIDF and PCA-CART.

Table 3.2 Classification accuracy, Precision and Recall using Camera Reviews (%)

<table>
<thead>
<tr>
<th>Classification Accuracy (%)</th>
<th>Precision (%)</th>
<th>Recall (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FE using TDFxIDF-NB</td>
<td>84.1</td>
<td>83.65</td>
</tr>
<tr>
<td>FE using TDFxIDF-CART</td>
<td>85.78</td>
<td>85.4033</td>
</tr>
<tr>
<td>FE using TDFxIDF-KNN</td>
<td>85.3</td>
<td>84.8767</td>
</tr>
<tr>
<td>FE using TDFxIDF and PCA-NB</td>
<td>85.18</td>
<td>84.7633</td>
</tr>
<tr>
<td>FE using TDFxIDF and PCA-CART</td>
<td>87.47</td>
<td>87.0833</td>
</tr>
<tr>
<td>FE using TDFxIDF and PCA-KNN</td>
<td>86.87</td>
<td>86.4633</td>
</tr>
<tr>
<td>FE using TDFxIDF and kernel PCA-NB</td>
<td>87.95</td>
<td>87.6067</td>
</tr>
<tr>
<td>FE using TDFxIDF and kernel PCA-CART</td>
<td>89.64</td>
<td>89.2933</td>
</tr>
<tr>
<td>FE using TDFxIDF and kernel PCA-KNN</td>
<td>89.16</td>
<td>88.7767</td>
</tr>
</tbody>
</table>
From Table 3.2 and Figure 3.5 it is observed that CART performs classification accuracy better than NB and KNN. Similarly FE using TDFxIDF and kernel PCA-CART performs better than FE using TDFxIDF-NB and FE using TDFxIDF-KNN. Results shows that the FE using TDFxIDF
and kernel PCA-CART performs accuracy better by 4.4% than FE using TDFxIDF-CART and by 2.45% than FE using TDFxIDF and PCA-CART.

From Table 3.2 and Figure 3.6 it is observed that CART performs precision better than NB and KNN. Similarly FE using TDFxIDF and kernel PCA-CART performs better than FE using TDFxIDF-NB and FE using TDFxIDF-KNN. Results shows that the FE using TDFxIDF and kernel PCA-CART performs precision better by 4.45% than FE using TDFxIDF-CART and by 2.5% than FE using TDFxIDF and PCA-CART.

**Figure 3.7 Recall for Camera Review**

From Table 3.2 and Figure 3.7 it is observed that CART performs recall better than NB and KNN. Similarly FE using TDFxIDF and kernel PCA-CART performs better than FE using TDFxIDF-NB and FE using TDFxIDF-KNN. Results shows that the FE using TDFxIDF and kernel PCA-CART performs recall better by 4.2% than FE using TDFxIDF-CART and by 2.3% than FE using TDFxIDF and PCA-CART.
3.4 CONCLUSION

A big role of information-collecting activity is to discover the thoughts of individuals. With accessibility as well as popularity of resources that are rich in opinions like online reviews, blog sites, more changes as well as obstacles rise because we are not capable of utilizing information technology to comprehend others' opinions. The efficacy of feature extraction methods and classification algorithms were investigated for classifying camera reviews. Reviews on camera are obtained from Amazon website. Feature from the reviews are extracted using TDF×IDF. Features are transformed using PCA and kernel PCA. Naïve Bayes and K-NN classifiers and CART algorithms classify the features as positive or negative. Experimental results demonstrate that features extracted using TDF×IDF with kernel PCA enhances the classification precision of the classifiers. Outcomes reveal that CART algorithm has higher classification accuracy than other classifiers.