Project Report

“Amazon Fine Food Review Analysis"

Introduction to the project

Project Title : Amazon Food Review Analysis

Project Duration : 8 Weeks

Project category : Machine Learning

Brief Description of project

Project Highlights:

This project is designed to give you a hands-on experience with natural language processing and work towards to develop a classification machine learning model on a real-world dataset. Many companies today collect vast amounts of data on customers reviews and clientele, and have a strong desire to understand the effect and result of their product so that they will improve their product. Being equipped with this information can assist a company engineer future products and services that best satisfy the demands or needs of their customers.

Data used: Kaggle Machine Learning Repository

Technology Used:

Language Used: Python3.

Framework Used: Jupyter Notebook

Operating System Used: Windows 10

Problem Statement:

This dataset consists of reviews of fine foods from amazon. The data span a period of more than 10 years, including all ~500,000 reviews up to October 2012. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories. Here our objective is to predict whether a review is **-ve(Rating 1 or 2) or +ve(Rating 4 or 5).**

Methodology Used

Supervised Learning: Supervised learning is the machine learning task of learning a function that maps an input to an output based on example input-output pairs. It infers a function from *labeled* training data consisting of a set of *training examples*. In supervised learning, each example is a *pair* consisting of an input object (typically a vector) and a desired output value (also called the *supervisory signal*).

Data Preparation: obtain dataset from internal and external sources.

Data consistency check in term of definition of fields , unit of measurements , time period etc.

Data Exploration and Conditioning: missing data handling ,outliers , graphical or visual analysis, Transformation ,Normalization, Partitioning into training , validation and test datasets.

Model Planning : Determining data mining task such as prediction , classification etc.

Select appropriate data modeling method such as KNN and Logistics Regression.

Model Building: Building different candidate model using selected techniques and their variant using training data.

Evaluate the final model on test data.

Model Interpretation: Model evaluation using key performance matrices.

Model deployment: Pilot project to integrate and run the model on the operational system.

Things you will learn by completing this project:

1. How to apply preprocessing techniques such as feature scaling and outlier detection.
2. How to apply Text preprocessing techniques such as lemmatization, stop-words and Stemming.
3. How to apply Natural language processing techniques such as Bag of words and TF-IDF.
4. Build Machine Learning Models.
5. Optimize the model to improve the accuracy

System Requirements

HARDWARE REQUIREMENTS

- Intel core i5(Processor)
- 8GB RAM
- 4MB Cache memory
- Hard Disk 50GB

SOFTWARE REQUIREMENTS

- Operating System: Windows xp ,Windows 7,Linux
- Programming language: Python3
- Web server: Wamp

Project Description:

Amazon fine food review dataset consists of reviews of fine foods from amazon. The data span a period of more than 10 years, including all ~500,000 reviews up to October 2012. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories.

Overview:

This Dataset contains 10 features/independent variables/predictors etc. We will look at the reviews of each and every customers and will analyse them using k-nearest neighbours algorithm.

- Id- row id
- Product Id- Unique identifier for the product
- User Id- Unique identifier for the user
- Profile Name -Profile name of user
- Helpfulness Numerator- Number of users who found the review helpful
- Helpfulness Denominator- Number of users who indicated whether they found the review helpful
- Score- Rating between 1 and 5
- Time – Time Stamp for the review
- Summary – Brief summary of Review
- Text – Text of review.

Objective:

objective is to predict whether a review is **-ve(Rating 1 or 2) or +ve(Rating 4 or 5).**

Table of Content:

1.Loading Dataset:

In This phase we load all the important libraries and our Raw data.

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	1	B001E4KFG0	A3SGXH7AUHU8GW	delmartian	1	1	positive	1303862400	Good Quality Dog Food	I have bought several of the Vitality canned d...
1	2	B00813GRG4	A1D87F6ZCVE5NK	dll pa	0	0	negative	1346976000	Not as Advertised	Product arrived labeled as Jumbo Salted Peanut...
2	3	B000LQOCH0	ABXLMWJIXXAIN	Natalia Corres "Natalia Corres"	1	1	positive	1219017600	"Delight" says it all	This is a confection that has been around a fe...
3	4	B000UA0QIQ	A395BORC6FGVXV	Karl	3	3	negative	1307923200	Cough Medicine	If you are looking for the secret ingredient i...
4	5	B006K2ZZ7K	A1UQRSCLF8GW1T	Michael D. Bigham "M. Wassir"	0	0	positive	1350777600	Great taffy	Great taffy at a great price. There was a wid...

2. Data Pre-processing: In this phase we try to remove the unwanted and erroneous data like outliers and duplicate data.

Data Cleaning: Deduplication

It is observed (as shown in the table below) that the reviews data had many duplicate entries. Hence it was necessary to remove duplicates in order to get unbiased results for the analysis of the data. Following is an example:

```
display= pd.read_sql_query("""
SELECT *
FROM Reviews
WHERE Score != 3 AND UserId="AR5J8UI46CURR"
ORDER BY ProductID
""", con)
display
```

	Id	ProductId	UserId	ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text
0	78445	B000HDL1RQ	AR5J8UI46CURR	Geetha Krishnan	2	2	5	1199577600	LOACKER QUADRATINI VANILLA WAFERS	DELICIOUS WAFERS. I FIND THAT EUROPEAN WAFERS ...
1	138317	B000HDOPYC	AR5J8UI46CURR	Geetha Krishnan	2	2	5	1199577600	LOACKER QUADRATINI VANILLA WAFERS	DELICIOUS WAFERS. I FIND THAT EUROPEAN WAFERS ...

3. **Text Pre-Processing:** Remove Stop-words

Finally Snowball Stemming the word (it was observed to be better In this phase we will pre-process the text data for further use in our model building some key techniques are as follows Stemming, stop-word removal and Lemmatization.

Now that we have finished deduplication our data requires some pre-processing before we go on further with analysis and making the prediction model.

Hence in the Pre-processing phase we do the following in the order below:-

1. Begin by removing the html tags
2. Remove any punctuations or limited set of special characters like , or . or # etc.
3. Check if the word is made up of English letters and is not alpha-numeric
4. Check to see if the length of the word is greater than 2 (as it was researched that there is no adjective in 2-letters)
5. Convert the word to lowercase
6. than Porter Stemming)

After which we collect the words used to describe positive and negative reviews

ProfileName	HelpfulnessNumerator	HelpfulnessDenominator	Score	Time	Summary	Text	CleanedText
G. Kleinschmidt	61	79	negative	2002-12-27	Great movie turned bad	Just to let you know, this movie is one of my ...	b'let know movi one person favorit ghost movi ...
Kazantzakis "hinterlands"	5	8	negative	2003-10-29	Woodstream Gopher Trap 0610	This is a poor excuse for a gopher trap. I hav...	b'poor excus gopher trap lot gopher use trap r...

4. Model Building: KNN with Bag of Words:

Here I will build the model using knn algorithm with bag of words technique of natural language processing.

KNN-

the ***k*-nearest neighbors algorithm** (***k*-NN**) is a non-parametric method used for classification and regression. In both cases, the input consists of the *k* closest training examples in the feature space. The output depends on whether *k*-NN is used for classification or regression:

- In *k-NN classification*, the output is a class membership. An object is classified by a plurality vote of its neighbors, with the object being assigned to the class most common among its *k* nearest neighbors (*k* is a positive integer, typically small)

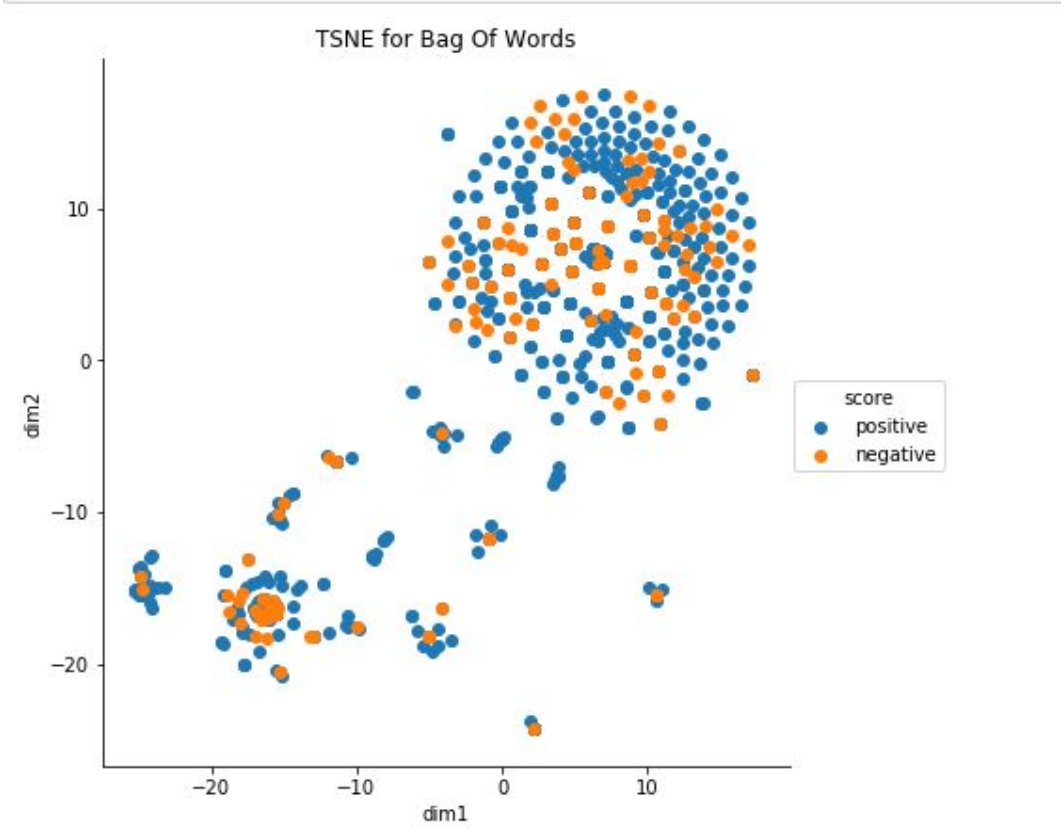
- If $k = 1$, then the object is simply assigned to the class of that single nearest neighbor.

- In *k-NN regression*, the output is the property value for the object. This value is the average of the values of k nearest neighbors.

Bag of Words:

The **bag-of-words model** is a simplifying representation used in natural language processing and information retrieval (IR). In this model, a text (such as a sentence or a document) is represented as the bag (multiset) of its words, disregarding grammar and even word order but keeping multiplicity.

Visualization For Bag of words:



Score prediction -KNN with Bag of word:

```
the accuracy score for bag of words model with optimal k=13 is 74.741667%
```

Confusion matrix:

A confusion matrix is a table that is often used to **describe the performance of a classification model** (or "classifier") on a set of test data for which the true values are known. The confusion matrix itself is relatively simple to understand, but the related terminology can be confusing.

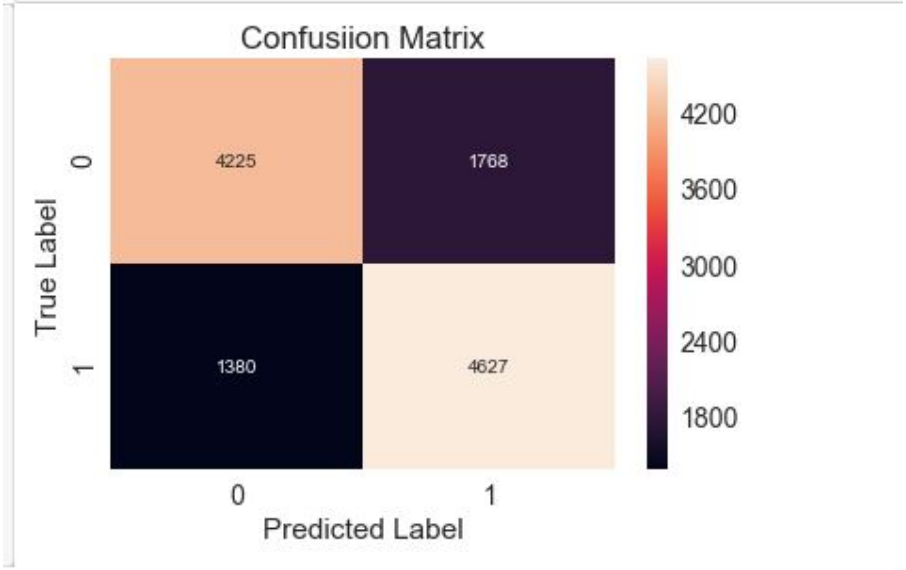
Let's now define the most basic terms, which are whole numbers (not rates):

- **true positives (TP):** These are cases in which we predicted yes (they have the disease), and they do have the disease.
- **true negatives (TN):** We predicted no, and they don't have the disease.
- **false positives (FP):** We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- **false negatives (FN):** We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

A couple other terms are also worth mentioning:

- **Null Error Rate:** This is how often you would be wrong if you always predicted the majority class. (In our example, the null error rate would be 60/165=0.36 because if you always predicted yes, you would only be wrong for the 60 "no" cases.) This can be a useful baseline metric to compare your classifier against. However, the best classifier for a particular application will sometimes have a higher error rate than the null error rate, as demonstrated by the Accuracy Paradox.
- **Cohen's Kappa:** This is essentially a measure of how well the classifier performed as compared to how well it would have performed simply by chance. In other words, a model will have a high Kappa score if there is a big difference between the accuracy and the null error rate.
- **F Score:** This is a weighted average of the true positive rate (recall) and precision.
- **ROC Curve:** This is a commonly used graph that summarizes the performance of a classifier over all possible thresholds. It is generated by plotting the True Positive Rate (y-axis) against the False Positive Rate (x-axis) as you vary the threshold for assigning observations to a given class.

Confusion matrix for KNN with Bag Of Words:

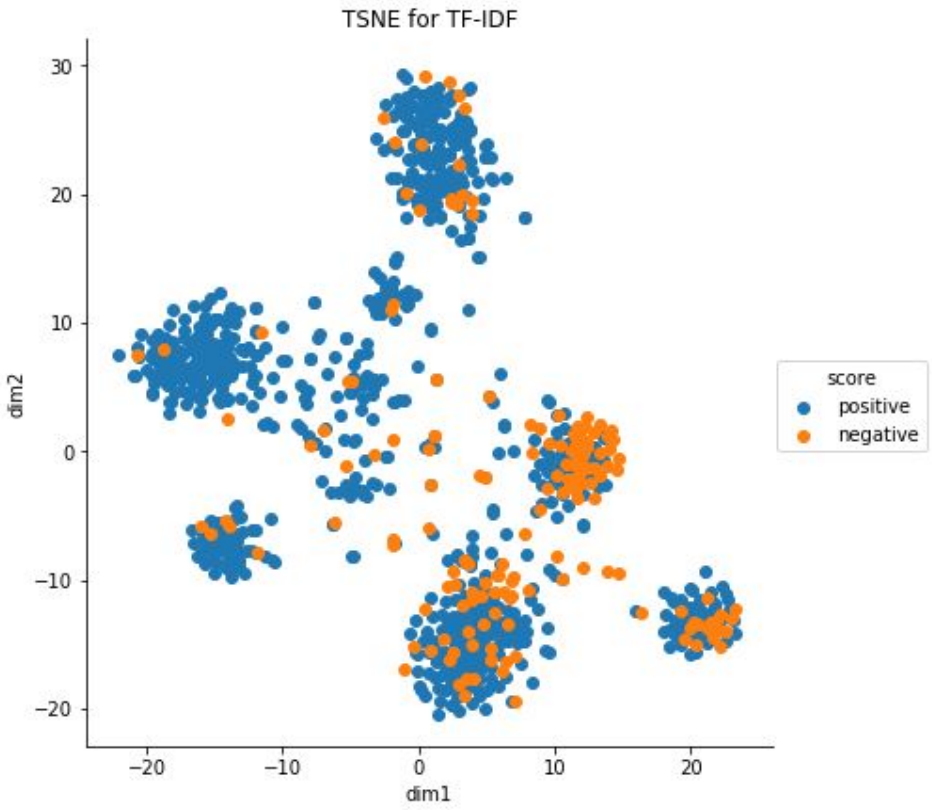


	precision	recall	f1-score	support
negative	0.75	0.75	0.75	5978
positive	0.75	0.75	0.75	6022
micro avg	0.75	0.75	0.75	12000
macro avg	0.75	0.75	0.75	12000
weighted avg	0.75	0.75	0.75	12000

TF-IDF:

In information retrieval, **tf–idf** or **TFIDF**, short for **term frequency–inverse document frequency**, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus. It is often used as a weighting factor in searches of information retrieval, text mining, and user modeling. The tf–idf value increases proportionally to the number of times a word appears in the document and is offset by the number of documents in the corpus that contain the word, which helps to adjust for the fact that some words appear more frequently in general. tf–idf is one of the most popular term-weighting schemes today; 83% of text-based recommender systems in digital libraries use tf–idf

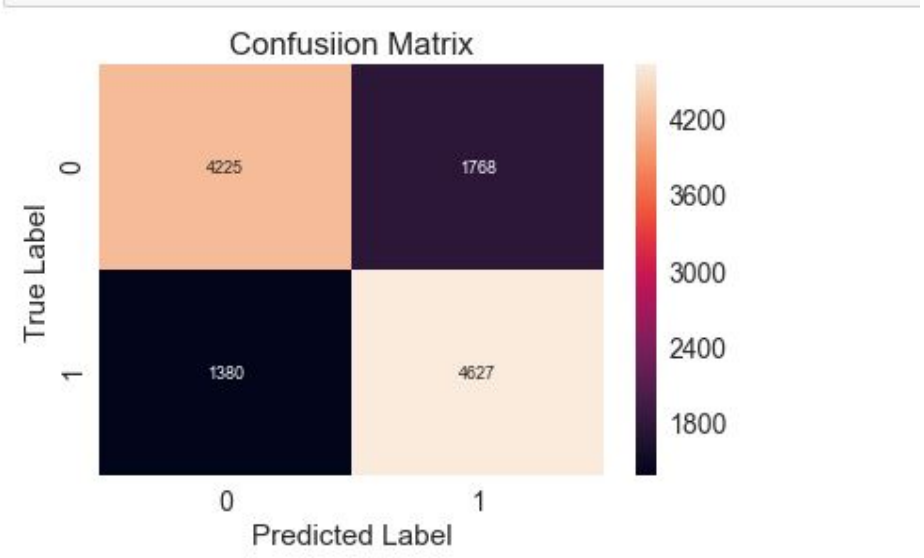
Visualization of TFIDF:



Score Prediction For KNN with TFIDF:

```
the accuracy score for bag of words model with optimal k=19 is 73.766667%
```

Confusion Matrix:



	precision	recall	f1-score	support
negative	0.75	0.70	0.73	5993
positive	0.72	0.77	0.75	6007
micro avg	0.74	0.74	0.74	12000
macro avg	0.74	0.74	0.74	12000
weighted avg	0.74	0.74	0.74	12000