(STAT 5870)
Project report on
Sentiment analysis of
Blog authorship corpus Data

By Sri Surya Sameer Vaddhiparthy (458704444)
Under the guidance of Dr. Kevin Lee

Introduction

Origin:
The blog authorship Corpus dataset contains a collection of posts of 19320 bloggers from blogger.com in 2004, a collection of 6,81,288 posts. There are bloggers from 3 age groups i.e. teens, adults and middle-aged people.

Columns:
Every data point includes the below attribute

- Blog ID
- Gender
- Age
- Topic
- Sign
- Date
- Text Content of the blog

```
In [15]: blog.columns
Out[15]: Index(['id', 'gender', 'age', 'topic', 'sign', 'date', 'text'], dtype='object')
```

Textual Content:
Along with text that makes sense, the text also contains blogger content that is filled with various undesired text like URLs, spelling mistakes and non-English language words, unwanted spacings and various symbols.

```
In [31]: blog["text"][388]
Out[31]: '             An attempt at Typographic Art...  Bloop!                  Bl00pS       blooop    blooop    blooop
blooP-a-dee-d00              '
```

Subset:
Due to limited computing power, I have considered a subset of the original data set with 50000 rows.

```
In [18]: blog.shape
Out[18]: (50000, 7)
```

Source:
The data set is available in csv format in the below link.
https://www.kaggle.com/rtatman/blog-authorship-corpus

Problem intended to solve:
It is practically impossible for a human to go through all the blogger content and summarize everything that is present, so I used text analytics techniques to clean and categorize the data into various age groups and then perform further analysis.

```
In [27]: blog["text"].head()
Out[27]:
0               Info has been found (+/- 100 pages,...
1               These are the team members:   Drewe...
2               In het kader van kernfusie op aarde...
3                     testing!!!  testing!!!
4                 Thanks to Yahoo!'s Toolbar I can ...
Name: text, dtype: object
```

Then use the cleaned version of the above data to understand and present the blogging patterns of people belonging to various age groups and genders in terms of graphical plots and word clouds.

Methodology

The libraries used in this project:

- Pandas
- Matplotlib.pyplot as plt,
- Nltk – Natural Language Toolkit
- Nltk.corpus for stopwords,
- Re (Regular expression Package)
- Wordcloud
- Textblob.

The methods used in the project listed under their primary objective:

1. **Reading and understanding the dataset.**
    a) Read 50,000 rows (specified a parameter for rows) - *pd.read_csv*
    b) Check presence of null values - *blog.isna().any*
    c) Summary of the blog - *blog.info*
    d) Remove rows with duplicate values (if any) - *blog.drop_duplicates*
    e) To get the number of rows and columns. - *blog.shape*
    f) View the first few rows - *blog.head*
    g) Get column names - *blog.columns*
    h) Data type of each column - *blog.dtypes*
    i) Get unique values from the specified column.
    blog.col_name.unique()
    j) Read only text content of the 111th row in the dataset.
    blog*["text"][111]*

2. **Cleaning of the text.**
    a) Dropping "ID" and "Date" columns.
    blog.drop(['id','date'], axis=1, inplace=True)
    b) Remove all numbers, symbols and extra spaces, keeping only upper case and lower case alphabets.
    blog['clean_text']=blog['text'].apply(lambda x: re.sub(r'[^A-Za-z]+',' ',x))
    c) Turn all text into lower case.
    blog['clean_text']=blog['clean_text'].apply(lambda x: x.lower())

d) Removing all spaces at the beginning and at the end of the text.
blog['clean_text']=blog['clean_text'].apply(lambda x: x.strip())

3. Word Elimination and deep cleaning of our text

a) Getting stopwords.
stopwords=set(stopwords.words('english'))
b) Removing stop words.
blog['clean_text']=blog['clean_text'].apply(lambda x: ' '.join([words for words in x.split() if words not in stopwords]))
c) Getting the list of English words.
words = set(nltk.corpus.words.words())
d) Remove non-English words and chat shortcut words that will not contribute to our analysis and get only clean English words with correct spellings.
blog['clean_text']=blog['clean_text'].apply(lambda x:' '.join(w for w in nltk.wordpunct_tokenize(x) \if w.lower() in words or not w.isalpha()))

4. **Categorizing the data for in-depth age-group-wise analysis. Defined below function that assigns age-groups based on age value in the column.**
def set_age_group(df): df["age_group"] = pd.cut(x=df['age'], bins=[10,20,30,40], labels=["Teens","Young-Adult","Middle-Aged"])
return df

5. Understanding the age distribution.

a) A mean age of the bloggers in my chosen subset.
blog['age'].mean()
b) Visualizing age distribution of bloggers in my Dataset.
sums = blog['age_group'].value_counts()
axis('equal');
pie(sums, labels=sums.index);
plt.title("Distribution of bloggers across age groups")
show()

6. **Text Analysis of the entire data.**
    a) To get polarity values
    TextBlob(clean_text).sentiment.polarity
    b) To get subjectivity values
    TextBlob(clean_text).sentiment.subjectivity

7. **Visualization of the results:**
    a) WordCloud
    wordcloud = WordCloud(height=2000, width=2000, stopwords=STOPWORDS, background_color='white').generate(all_words_t_student)
    b) Distribution Plot.
    sns.distplot
    c) Boxplot.
    sns.boxplot

Results

1. **Reading the last rows:**

   This is a sample of text from the last few rows with various shortcut English words and unwanted punctuations.

```
49995          Aug 7th Thur... Bought Her Mua Chee & S...
49996          Aug 6th Wed.. Her 1st Day @ Work Back @...
49997          Aug 4th Mon Zing's BD !! Went To Her Pl...
49998          Aug 3rd Sun.. Went To Her Place B4 Goin...
49999          Aug 1st Fri.. Met Her To Go Shoppin' @ ...
Name: text, dtype: object
```

2. **After cleaning the text, stop words removal.**

   The below before and after text samples demonstrate the text cleaning.

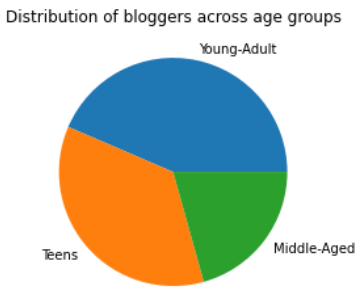
   A Portion of the original text (row 10)

" Ah, the Korean language...it looks so difficult at first, then as you figure out how to read Hanguel (Korea's surprisingly easy-to-learn alphabet of 24 characters) it seems so easy. Then the vocabulary starts. Oh no. Then the backwards (to us) sentence structure. Yikes! Luckily there are many options for us slow-witted foreigners to take on the language. Of course I could list them here but urlLink this JoongAng article says a lot and there are more resources urlLink here . Well, if you're a guy here is some motivation for you: Jeon Ji Hyun (전지현)

   Cleaned text (row 10)

'ah language difficult first figure read surprisingly easy learn alphabet easy vocabulary oh backwards us sentence structure luckily many us slow witted take language course could list article lot well guy motivation latest something actually star hear commercial feature positive saw latest movie night hard describe name version version short ne like introduce surprisingly make sense like quite good actually movie shown special times list many click great reason learn already married went well local national course take picture put movie bar update bud mine link ad apparently nothing sort'

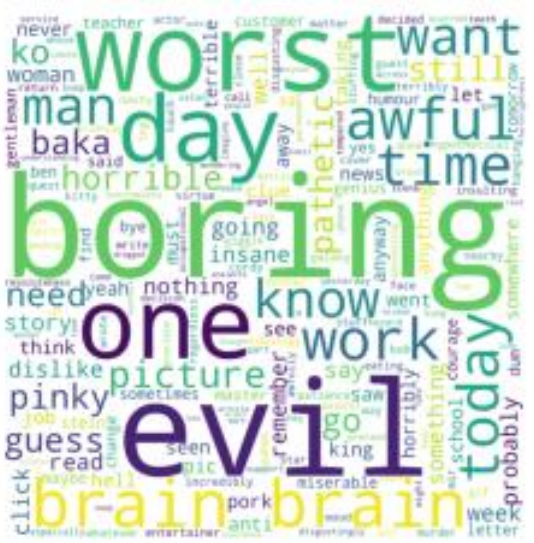
3. **After categorizing data based on age for a new column age_group**



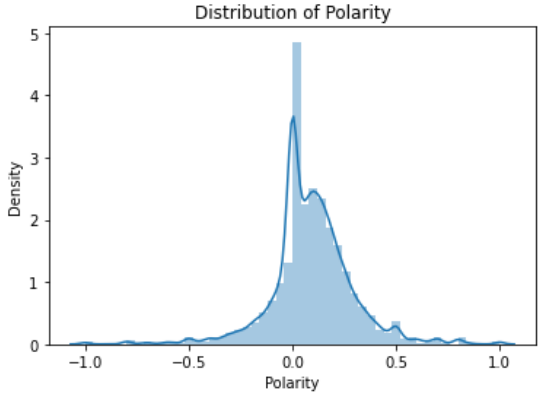
	age	age_group
0	15	Teens
1	15	Teens
2	15	Teens
3	15	Teens
4	33	Middle-Aged

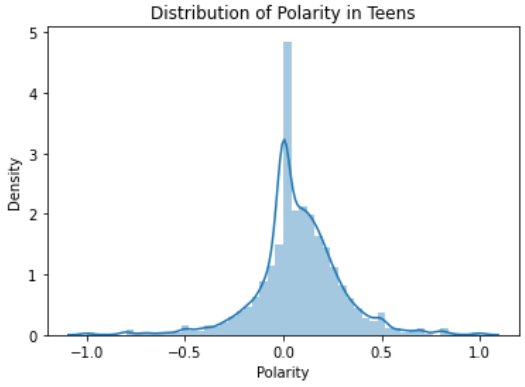
4. **The Polarity of the blog posts.**

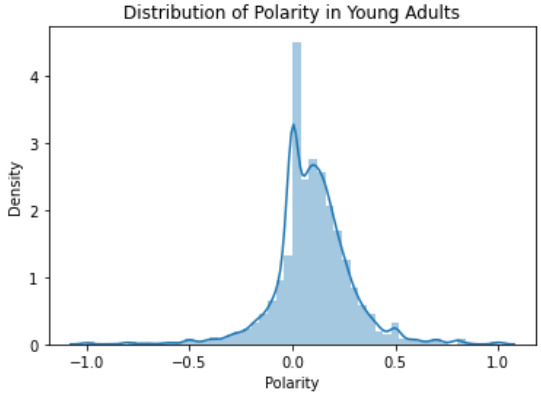
We can visualize the words that make a blog posts to be classified as the most positive or most negative ones with help of below Word Clouds.

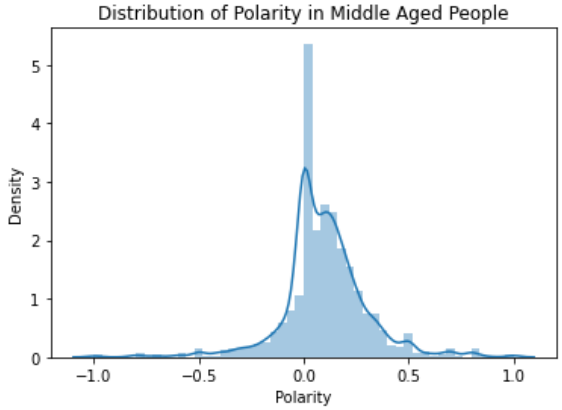


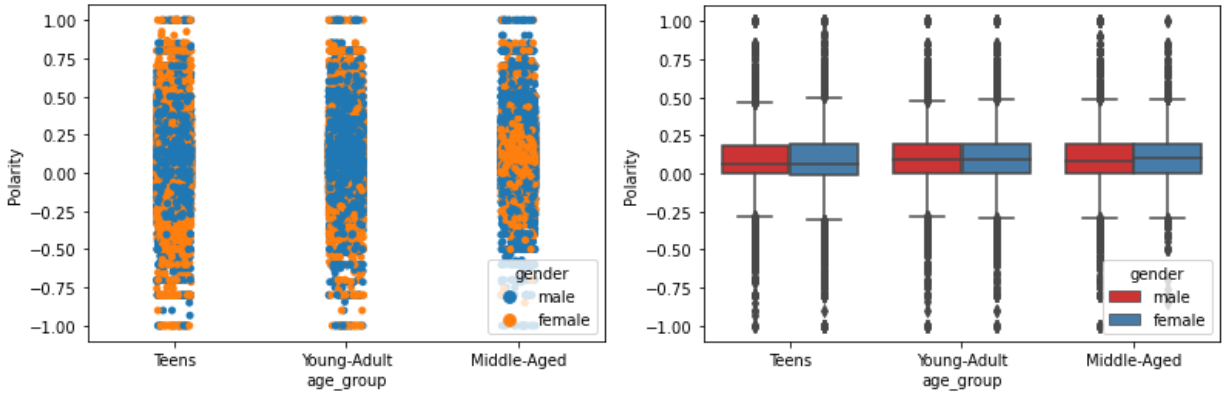
The below figures demonstrate the statistical distribution of polarity across overall data and then respective age groups.











age_group	
Teens	0.078061
Young-Adult	0.099220
Middle-Aged	0.100101

The statistical observations of polarity across various age groups suggest an overall positive attitude of bloggers in their blog posts with a few outliers on both sides.

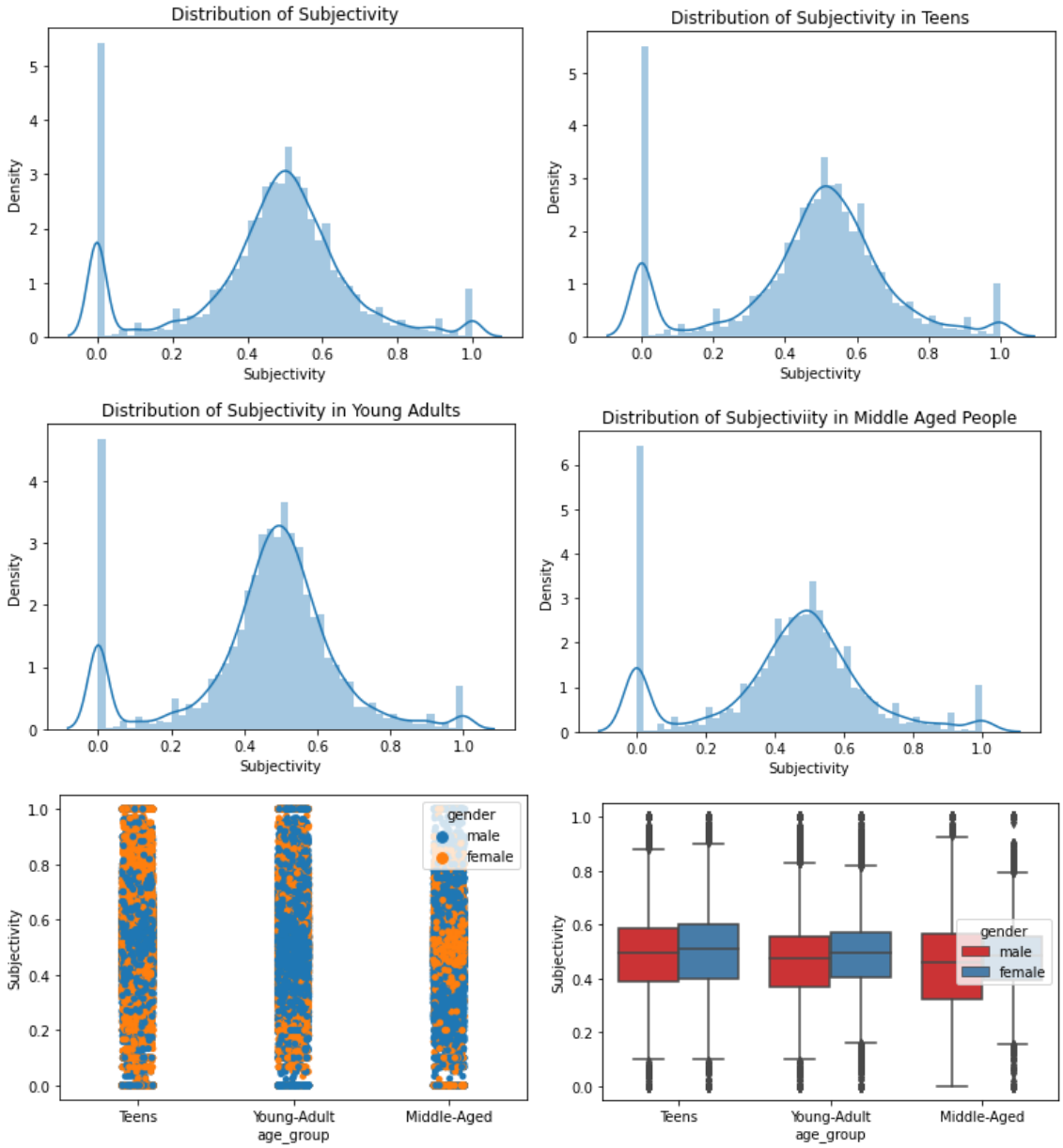
5. **The Subjectivity of the blog posts**

We can visualize the words that make a blog post to be classified as most subjective or least subjective with help of the below Word Clouds.

The following figures demonstrate the statistical distribution of subjectivity across overall data and then respective age groups.

age_group	
Teens	0.472132
Young-Adult	0.457551
Middle-Aged	0.437575

The mean statistical observations of subjectivity across various age groups are tabulated in the adjacent image.



6. Visualization using Word Cloud

Topic-Based:

Some interesting trends in word cloud based on the topic, the student topics have frequently used words like “think” and “well”, the religion topic has “god” and “one” as the frequently used words, which correlates with a common intuition.

Age group based: The word “time” starts to appear more frequently in blogs by middle age and young adult age groups and the word “think” in the teenage group conveying the mindset and thought process variation.

Topic-based word clouds

Age-based word cloud
(Words not filtered)

Age-based word cloud
(Common words filtered)

I observed the below trends in Word Cloud when we go from middle age group to young-adult to teens for the below words:

New: *The usage decreases.*

Cold: *The usage decreases.*

Back: *The usage decreases slightly.*

Conclusion

I have successfully

- ✓ Extracted the subset of a real-world text dataset called Blog authorship corpus.
- ✓ Extracted the useful and clean text.
- ✓ Analyzed statistically using plots and visualized the frequently used words.
- ✓ Identified and presented the trends in the variation of their usage.

Future Scope

- ✓ The code can be applied to a large amount of data if enough computing power is present to yield more statistically significant insights.
- ✓ My work can be used to identify the negative blog posts that spread negativity and can be taken down automatically, which would otherwise take hours to review manually.
- ✓ Also, we can understand the trend in usage of words across various other blog posts and its fluctuations across various demographics.
- ✓ Further work on the topic modeling with data will help us classify the data based to content and present relevant content to the users.