TAR UC E-DATA HACKATHON 2020

C11: TEAM FALCON

TUNKU ABDUL RAHMAN UNIVERSITY COLLEGE

Team Leader : Lim Jun Rong

Member 1 : Lai Xin Yi

Member 2 : Ong T'nsam

Business Understanding

E-COMMERCE

- Electronically buying or selling of products

- Through Online

- Eg. a) Clothing
  b) Seafood
  c) Accessories

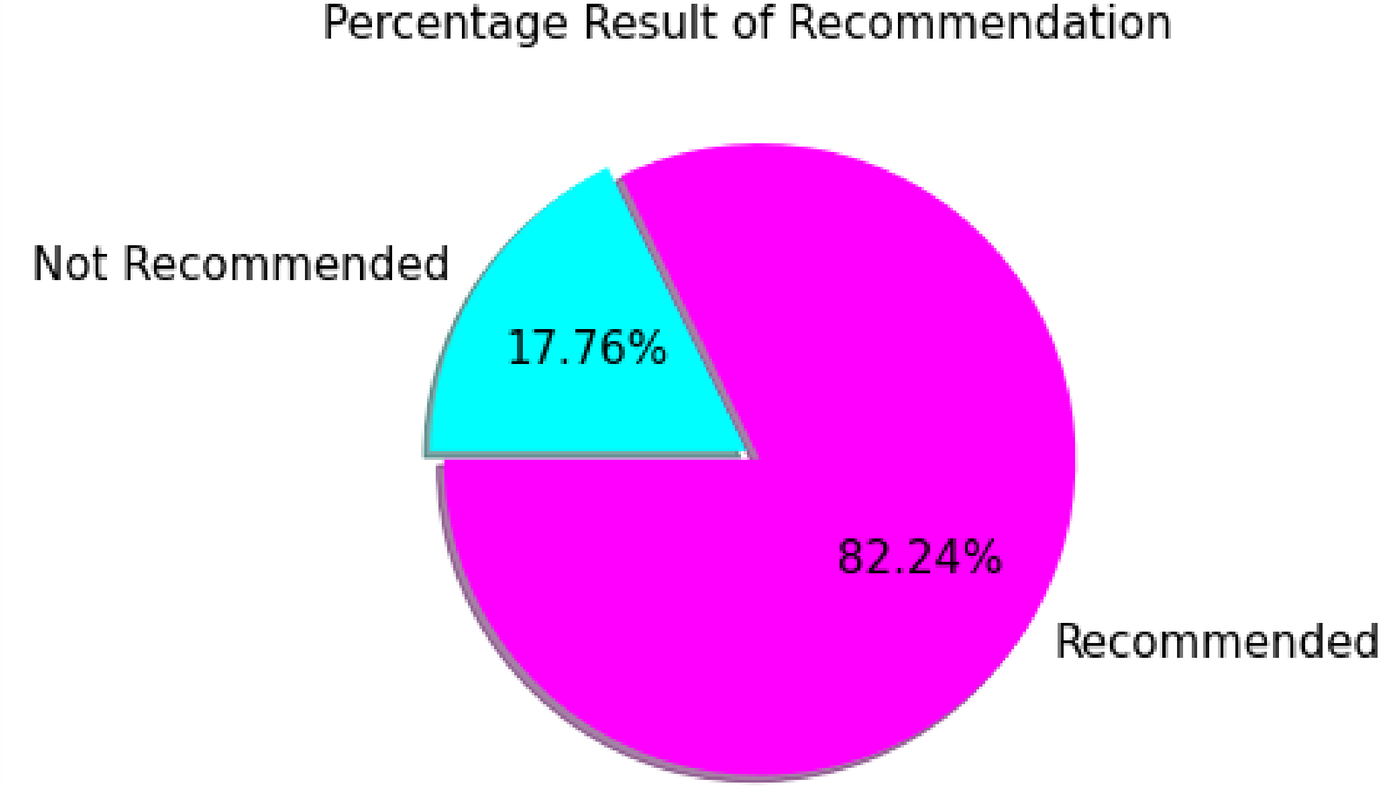
Data Understanding

- 10 columns and 23,485 rows
- Features:

1. Clothing_ID
2. Age
3. Title
4. Review_Text ✓
5. Rating
6. Recommended_IND ✓
7. Positive_Feedback_Count
8. Division_Name
9. Department_Name
10. Class_Name

Data Understanding (cont.)

```
Recommended          19314
Not Recommended       4172
Name: Recommended_IND, dtype: int64
```



Data Preprocessing

1. Remove Unwanted Columns

```
# Removing unwanted columns
df = df.loc[:, df.columns.intersection(['Review_Text','Recommended_IND'])]
print(f'Updated Shape of Data: {df.shape}')
```

```
Updated Shape of Data: (23486, 2)
```

	Review_Text	Recommended_IND
0	Absolutely wonderful - silky and sexy and comf...	1
1	Love this dress! it's sooo pretty. i happene...	1
2	I had such high hopes for this dress and reall...	0
3	I love, love, love this jumpsuit. it's fun, fl...	1
4	This shirt is very flattering to all due to th...	1

Data Preprocessing

2. Handle Missing Value

Review_Text	True	845
Recommended_IND	False	0

```
t_cells = np.product(df.shape)
t_missing = df.isnull().sum().sum() # total missing value for all the varible
# percent of data that is missing
percent_missing = (t_missing/t_cells) * 100
print("Percentage of missing value: {:.2f}%".format(percent_missing))
```

```
Percentage of missing value: 1.80%
```

Data Preprocessing

2. Handle Missing Value (Cont.)

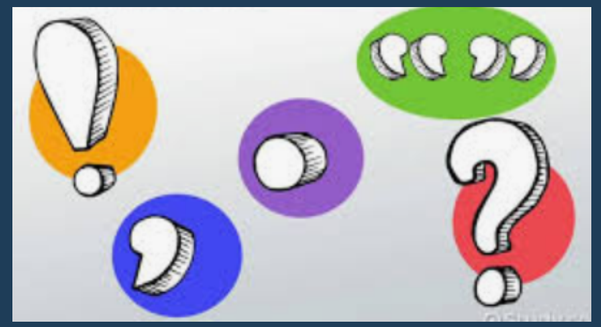
```
#double comfirm missing value
nullValue = df.isnull().any()
nullValue
```

```
Review_Text          False
Recommended_IND      False
```

Data Preprocessing

3. Remove Punctuation & Convert to Lower-case

```
def text_clean(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)
    text = re.sub('\w*\d\w*', '', text)
    text = re.sub('[‘’“”…]', '', text)
    text = re.sub('\n', '', text)
    text= text.strip()

    return text

cleaned = lambda x: text_clean(x)
```

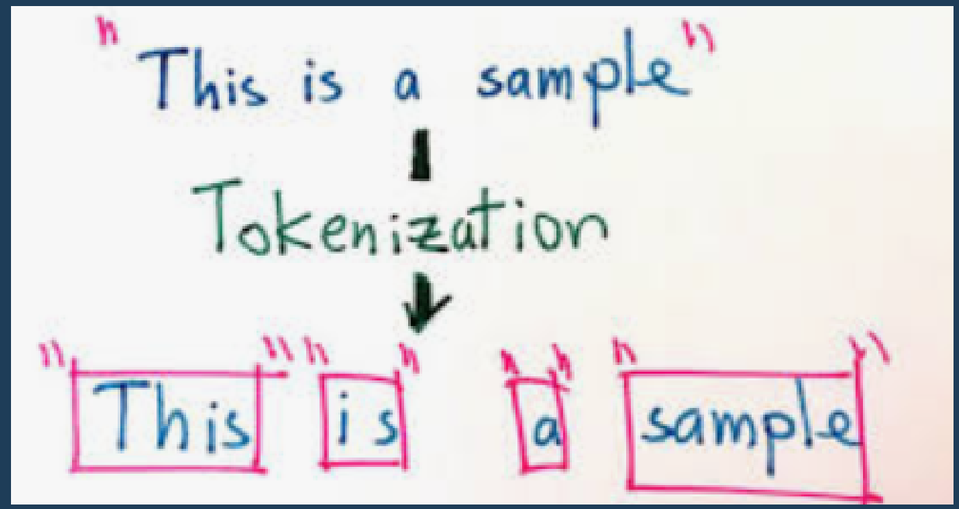


3. Remove Punctuation & Convert to Lower-case (Cont.)

	Review_Text	Recommended_IND	New_Review_Text
0	Absolutely wonderful - silky and sexy and comf...	1	absolutely wonderful silky and sexy and comfo...
1	Love this dress! it's sooo pretty. i happene...	1	love this dress its sooo pretty i happened t...
2	I had such high hopes for this dress and reall...	0	i had such high hopes for this dress and reall...
3	I love, love, love this jumpsuit. it's fun, fl...	1	i love love love this jumpsuit its fun flirty ...
4	This shirt is very flattering to all due to th...	1	this shirt is very flattering to all due to th...
5	I love tracy reese dresses, but this one is no...	0	i love tracy reese dresses but this one is not...
6	I aded this in my basket at hte last mintue to...	1	i aded this in my basket at hte last mintue to...
7	I ordered this in carbon for store pick up, an...	1	i ordered this in carbon for store pick up and...
8	I love this dress. i usually get an xs but it ...	1	i love this dress i usually get an xs but it r...
9	I'm 5"5' and 125 lbs. i ordered the s petite t...	1	im and lbs i ordered the s petite to make su...

Data Preprocessing

4. Tokenisation




```
def tokens(text):
    tokens = nltk.word_tokenize(str(text))
    # taken only words (not punctuation)
    token_words = [w for w in tokens if w.isalpha()]
    return token_words

cleaned1 = lambda x: tokens(x)
```

Data Preprocessing

4. Tokenisation (Cont.)

	Review_Text	Recommended_IND	New_Review_Text
0	Absolutely wonderful - silky and sexy and comf...	1	[absolutely, wonderful, silky, and, sexy, and,...
1	Love this dress! it's sooo pretty. i happene...	1	[love, this, dress, its, sooo, pretty, i, happ...
2	I had such high hopes for this dress and reall...	0	[i, had, such, high, hopes, for, this, dress, ...
3	I love, love, love this jumpsuit. it's fun, fl...	1	[i, love, love, love, this, jumpsuit, its, fun...
4	This shirt is very flattering to all due to th...	1	[this, shirt, is, very, flattering, to, all, d...
5	I love tracy reese dresses, but this one is no...	0	[i, love, tracy, reese, dresses, but, this, on...
6	I aded this in my basket at hte last mintue to...	1	[i, aded, this, in, my, basket, at, hte, last,...
7	I ordered this in carbon for store pick up, an...	1	[i, ordered, this, in, carbon, for, store, pic...
8	I love this dress. i usually get an xs but it ...	1	[i, love, this, dress, i, usually, get, an, xs...
9	I'm 5"5' and 125 lbs. i ordered the s petite t...	1	[im, and, lbs, i, ordered, the, s, petite, to,...

Data Preprocessing

5. Remove Stopwords

```
stop_words = set(stopwords.words('english'))

def remove_stopwords(text):
  remove = [w for w in text if w not in stop_words]
  return remove

cleaned2 = lambda x: remove_stopwords(x)
```

	Review_Text	Recommended_IND	New_Review_Text
0	Absolutely wonderful - silky and sexy and comf...	1	[absolutely, wonderful, silky, sexy, comfortable]
1	Love this dress! it's sooo pretty. i happene...	1	[love, dress, sooo, pretty, happened, find, st...
2	I had such high hopes for this dress and reall...	0	[high, hopes, dress, really, wanted, work, ini...
3	I love, love, love this jumpsuit. it's fun, fl...	1	[love, love, love, jumpsuit, fun, flirty, fabu...
4	This shirt is very flattering to all due to th...	1	[shirt, flattering, due, adjustable, front, ti...

Data Preprocessing

6. Lemmetisation

a) Verb - Remove "ing", "s", and "ed"

```
# Lemmatize the verb
def lem1(text):
    wordnet = WordNetLemmatizer()
    lemma_words = []
    for w in text:
      lemma_words.append(wordnet.lemmatize(w,'v'

    return lemma_words

cleaned3 = lambda x: lem1(x)
```

sleeping, slept, sleeps --> sleep

moves, moving , moved --> move

eats, eating --> eat

Data Preprocessing

b) Noun

problems --> problem
laptops --> laptop
friends --> friend

```
# Lemmatize the nouns
def lem2(text):
    wordnet = WordNetLemmatizer()
    lemma_words = []
    for w in text:
      lemma_words.append(wordnet.lemmatize(w,'n'))

    return lemma_words

cleaned4 = lambda x: lem2(x)
```

Data Preprocessing

6. Lemmetisation

	Review_Text	Recommended_IND	New_Review_Text
0	Absolutely wonderful - silky and sexy and comf...	1	[absolutely, wonderful, silky, sexy, comfortable]
1	Love this dress! it's sooo pretty. i happene...	1	[love, dress, sooo, pretty, happen, find, stor...
2	I had such high hopes for this dress and reall...	0	[high, hop, dress, really, want, work, initial...
3	I love, love, love this jumpsuit. it's fun, fl...	1	[love, love, love, jumpsuit, fun, flirty, fabu...
4	This shirt is very flattering to all due to th...	1	[shirt, flatter, due, adjustable, front, tie, ...
5	I love tracy reese dresses, but this one is no...	0	[love, tracy, reese, dress, one, petite, foot,...
6	I aded this in my basket at hte last mintue to...	1	[aded, basket, hte, last, mintue, see, would, ...
7	I ordered this in carbon for store pick up, an...	1	[order, carbon, store, pick, ton, stuff, alway...
8	I love this dress. i usually get an xs but it ...	1	[love, dress, usually, get, x, run, little, sn...
9	I'm 5"5' and 125 lbs. i ordered the s petite t...	1	[im, lb, order, petite, make, sure, length, wa...

Data Preprocessing

7. Remove Non-English Words

```
dictionary = enchant.Dict("en_US")

def remove_non_english(text):
  remove = []
  for w in text:
    if dictionary.check(w):
      remove.append(w)

  return remove

cleaned5 = lambda x: remove_non_english(x)
```

Data Preprocessing

7. Remove Non-English Words (Cont.)

	Review_Text	Recommended_IND	New_Review_Text
0	Absolutely wonderful - silky and sexy and comf...	1	[absolutely, wonderful, silky, sexy, comfortable]
1	Love this dress! it's sooo pretty. i happene...	1	[love, dress, pretty, happen, find, store, gla...
2	I had such high hopes for this dress and reall...	0	[high, hop, dress, really, want, work, initial...
3	I love, love, love this jumpsuit. it's fun, fl...	1	[love, love, love, jumpsuit, fun, flirty, fabu...
4	This shirt is very flattering to all due to th...	1	[shirt, flatter, due, adjustable, front, tie, ...
5	I love tracy reese dresses, but this one is no...	0	[love, dress, one, petite, foot, tall, usually...
6	I aded this in my basket at hte last mintue to...	1	[basket, last, see, would, look, like, person,...
7	I ordered this in carbon for store pick up, an...	1	[order, carbon, store, pick, ton, stuff, alway...
8	I love this dress. i usually get an xs but it ...	1	[love, dress, usually, get, x, run, little, sn...
9	I'm 5"5' and 125 lbs. i ordered the s petite t...	1	[lb, order, petite, make, sure, length, long, ...

Sentiment Analysis

Word Cloud

```
def wordcloud_draw(data, color = 'black'):
    words = ' '.join(str(data))

    wordcloud = WordCloud(background_color=color,
                          width=2500,
                          height=2500
                          ).generate(str(data))
    plt.figure(1,figsize=(50, 10))
    plt.imshow(wordcloud)
    plt.axis('off')
    plt.show()
```

```
[100] print("Positive words")
      wordcloud_draw(recommended,'white')
      print("Negative words")
      wordcloud_draw(not_recommended)
```

Sentiment Analysis (cont.)

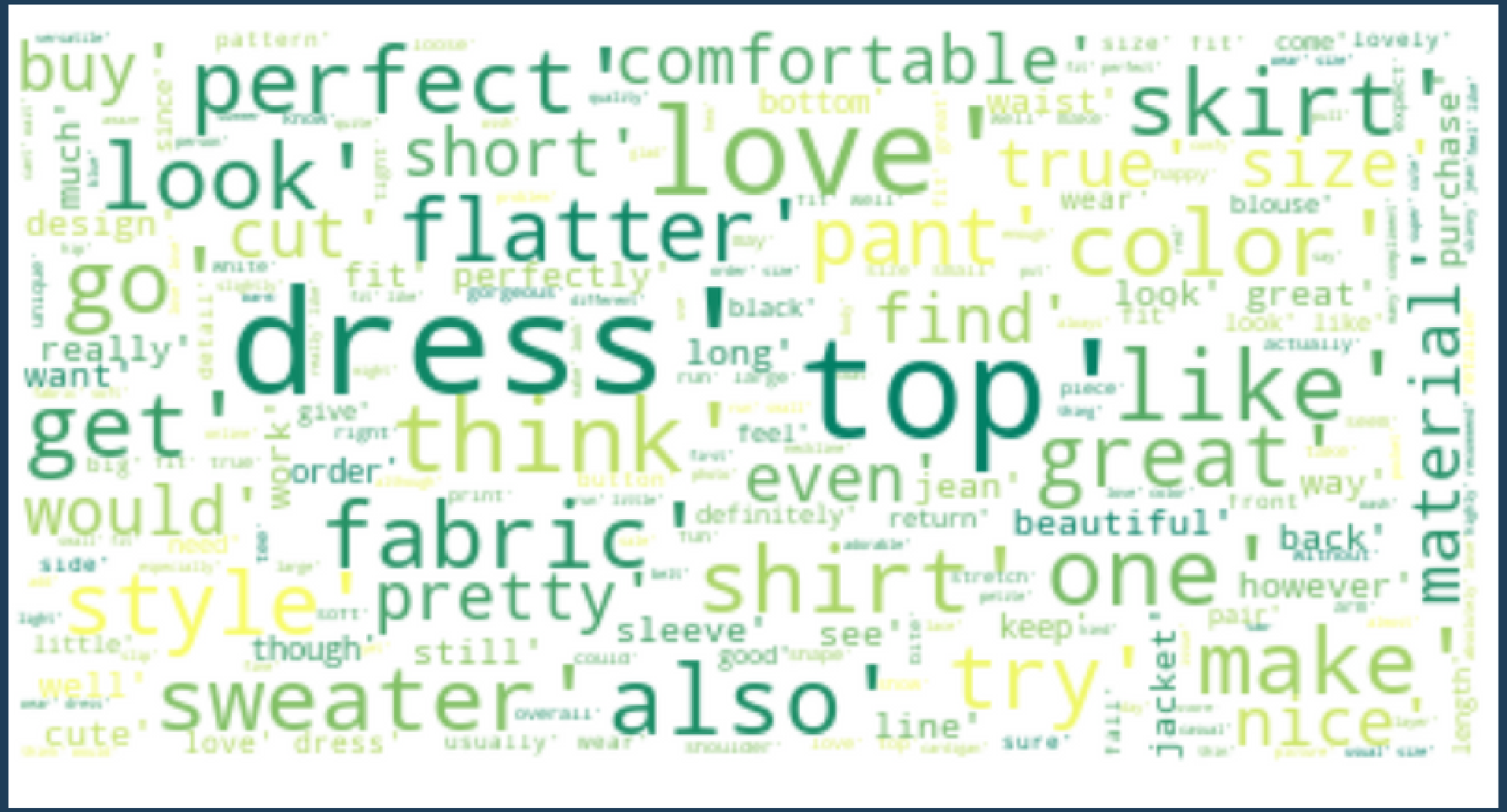
Word Cloud - Sample 1

Positive Comment

Negative Comment

Word Cloud - Sample 2

Positive Comment



Negative Comment

Feature Generation using Bag-of-words

	ab	abbey	abdomen	abdominal	abhor	ability	abject	able	abnormal	abnormally	abroad	abruptly	absence	absolu
0	0	0	0	0	0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
22636	0	0	0	0	0	0	0	0	0	0	0	0	0	
22637	0	0	0	0	0	0	0	0	0	0	0	0	0	
22638	0	0	0	0	0	0	0	1	0	0	0	0	0	
22639	0	0	0	0	0	0	0	0	0	0	0	0	0	
22640	0	0	0	0	0	0	0	0	0	0	0	0	0	

22641 rows × 6953 columns

Train-Test Data Split

Purpose : To split data into 2 parts which are testing and training data

```
X = bag_of_words
y = df['Recommended_IND']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
print("Training Shape: ",X_train.shape)
print("Testing Shape: ",X_test.shape)
```

```
Training Shape:  (18112, 6953)
Testing Shape:  (4529, 6953)
```

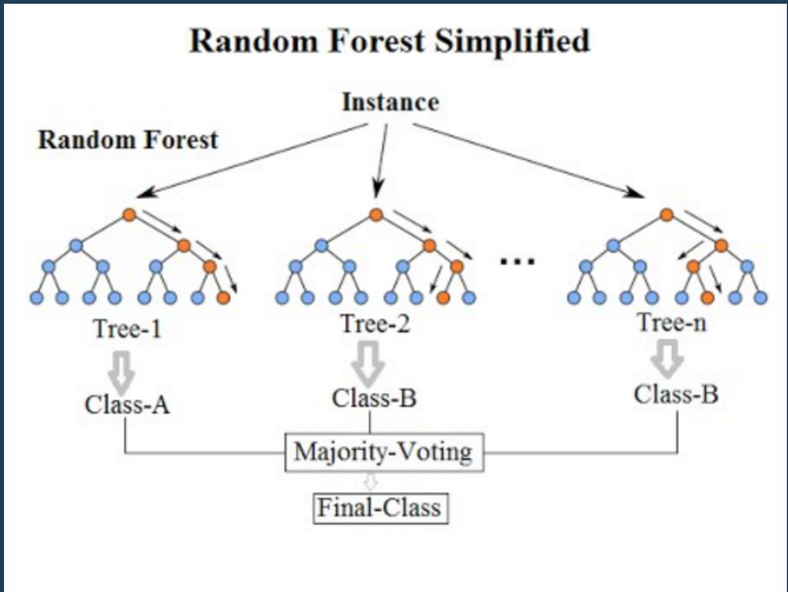
SMOTE

Purpose : Handle imbalance data.

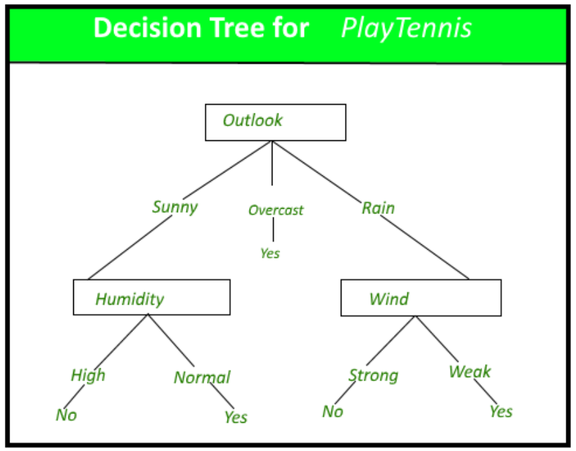
```
Before OverSampling, counts of label '1': 14823
Before OverSampling, counts of label '0': 3289

After OverSampling, the shape of train_X: (29646, 6953)
After OverSampling, the shape of train_y: (29646,)

After OverSampling, counts of label '1': 14823
After OverSampling, counts of label '0': 14823
```

Data Modelling

1. Random Forest

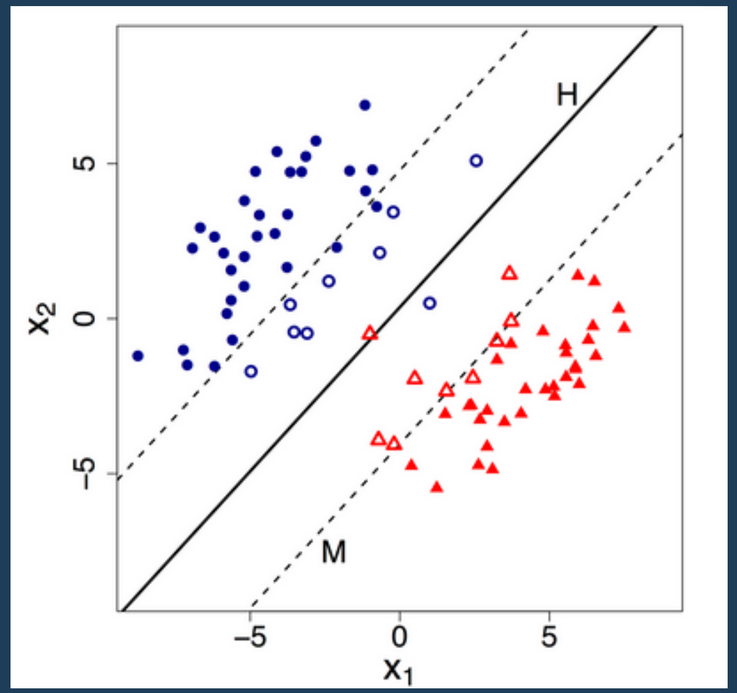


2. Decision Tree

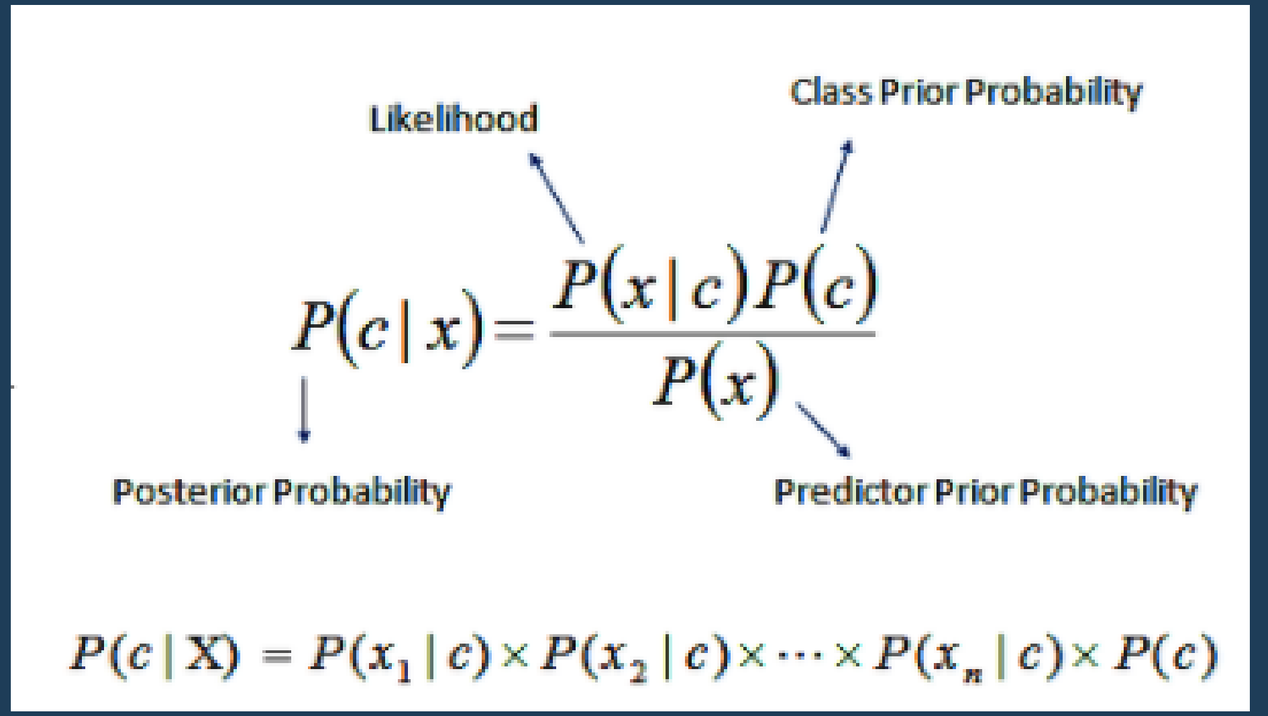


Data Modelling (cont.)

3. Support Vector Machine (SVM)



4. Naive Bayes



Data Evaluation

Performance metrics

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Precision 

False positive rate 

Comparison between performance metrics

Model	Accuracy_score	Recall_score	Precision	f1_score
Random Forest	0.827114	0.935701	0.864744	0.898824
Decision Tree	0.769265	0.835082	0.877828	0.855922
SVM	0.866416	0.938391	0.902692	0.920195
Naive Bayes	0.862884	0.884046	0.945339	0.913666

Test Customer's Input

https://colab.research.google.com/drive/1wY0U2Ek1agqvYyo77jcbE7_cudp3h9Wy?usp=sharing

Github Link

https://github.com/JunRong00/TAR-UC-E-Data-Hackathon-2020


THANK
YOU