GROUP 5

Topic: Lazada product recommendation

- Machine Learning

DUONG THI HUE
20176772

GIAP THI THUY VAN
20176905
- **LEADER**

TRAN THI HANG
20176748

TA DINH SON
20176862

Problem

Product's link in Lazada

⟶ Predict product is good, neutral or bad, base on comments of reviewers

⟶ Give advice whether to buy or not

Input:
Product's link in Lazada

Output:
Rank of product in range from 0 to 2 and recommendation for customer

Data scraping

- Scrape comments from Lazada by using ***selenium***

```
def load_url_selenium (url):
    driver=webdriver.Chrome(executable_path='/usr/bin/chromedriver')
    print("Loading url=", url)
    driver.get(url)
    review_csv=[]
    while True:
        #Get the review details here
        WebDriverWait(driver,10).until(EC.visibility_of_all_elements_located((By.CSS_SELECTOR,"div.item")))
        product_reviews = driver.find_elements_by_css_selector("[class='item']")
        # Get product review
        for product in product_reviews:
            review = product.find_element_by_css_selector("[class='content']").text
            if (review != "" or review.strip()):
                print(review, "\n")
                review_csv.append(review)
        #Check for button next-pagination-item have disable attribute then jump from loop else click on the next button
        if len(driver.find_elements_by_css_selector("button.next-pagination-item.next[disabled]"))>0:
            break;
        else:
            button_next=WebDriverWait(driver, 10).until(EC.visibility_of_element_located((By.CSS_SELECTOR, "button.next-pagination-item.next")))
            driver.execute_script("arguments[0].click();", button_next)
            print("next page")
            time.sleep(2)
    driver.close()
    print(review_csv)
    return review_csv
```

- Label training data set

stars	comment
2	ok
0	máy mới sạc chưa sài mà đã hết pin .cảm ứng rất chậm
1	chụp hình hơi xấu,máy chậm thua samsung lâu hết pin

Data pre-processing

Raw data are comments include:

- Special characters
- Discrete words
- Repeated and insignificant words

Solution →

- Remove special characters
- Combine discrete words into significant words
- Calculate frequency and importance of words

Data cleaning

- Remove special characters (comma, colon, semicolon, exclamation mark, etc.):

Regex (module ***re*** in Python)

```
# Standardize text data
def standardize_data(row):
    # Delete dot, comma, question mark at the end of sentence
    row = re.sub(r"[\.,\?]+$-", "", row)
    # Delete dot, comma, semicolon, colon, etc. in the sentence
    row = row.replace(",", " ").replace(".", " ") \
        .replace(";", " ").replace("“", " ") \
        .replace(":", " ").replace("”", " ") \
        .replace('"', " ").replace("'", " ") \
        .replace("!", " ").replace("?", " ") \
        .replace("-", " ").replace("?", " ")
    row = row.strip()
    return row
```

Data integrating

- Combine discrete words into meaningful words:

  word_tokenize from ***underthesea*** (Vietnamese NPL)

```
# Tokenizer
def tokenizer(row):
    return word_tokenize(row, format="text")
```

```
data_frame[0] = data_frame[0].apply(tokenizer)
```

Data pre-processing

Using ***BERT*** - Natural Language Processing pre-training

- Calculate frequency and importance of words:

  TfidfVectorizer from ***sklearn***

- Convert into numeric vector

  Word2Vec

⟶ Result:

```
After preprocess data:
Đã nhận được hàng máy đẹp giao hàng nhanh Cảm_ơn lazada


After embedding data:
  (0, 12)       0.6367254536854099
  (0, 7)        0.7710905891197927
```

Machine Learning techniques

Classification

- Response variable: Categorical
- Model: Supervised
- Objective: Predict

- · **K nearest-neighbour**

  - k = 5, p = 1

```
lr_clf = neighbors.KNeighborsClassifier(n_neighbors = 5, p = 1)
lr_clf.fit(train_features, train_labels)
```

- · **Bernoulli Naive Bayes**

  - X = {x_1, x_2, x_3, ...}, calculate probability of each class

$$\max_{C_i \in C} \left(P(C_i) \prod_{k=1}^{n} P(x_k | C_i) \right)$$

```
lr_clf = BernoulliNB(binarize = .5)
lr_clf.fit(train_features, train_labels)
```

Model evaluation and selection

K nearest-neighbour

- Small complexity
- Have to save all data in memory
- Result might not right when k is small and noise
- Doesn't require any **training**

Bernoulli Naive Bayes

- Faster when applied to big data
- Require training

	K-nn	Bernoulli Naive Bayes
Accurency	0.85	0.72

QUESTIONS & ANSWERS