به نام حق

گزارش پروژه پایانی در بازیابی اطلاعات و جستجو وب علیرضا پرچمی ۹۴۳۶۱۸۱۱۳۰۰۴

Text Classification جهت پیش‌بینی label سندها در Data mining و Information Retrieval استفاده می‌شود. در این پروژه از ۳ کتابخانه Scikit-Learn (sklearn) جهت آموزش و دسته‌بند کردن داکیومنت‌ها، Pandas جهت کار کردن بهتر و راحت‌تر با داده‌ها و nltk جهت پیش‌پردازش داده‌ها که شامل حذف StopWordها و Stemming استفاده شده‌است.

در شکل زیر، توابع read_file و preprocessing جهت خواندن و پیش‌پردازش سندها استفاده شده‌است:

```
def read_file():
    data = pd.read_csv('./Document/SMSSpamCollection', sep='\t')
    return data


def preprocessing():
    # Word Tokenization
    data['text'] = data['text'].apply(word_tokenize)
    # Remove Stopwords
    stopWords = set(stopwords.words('english'))
    data['text'] = data['text'].apply(lambda x: [item for item in x if item not in stopWords])
    # Stemming
    ps = PorterStemmer()
    data['text'] = data["text"].apply(lambda x: [ps.stem(y) for y in x])
```

در تابع ngrams که سه بار با ورودی های {۱و۲و۳} خوانده می‌شود، ابتدا n-gramها تولید می‌شود و سپس با استفاده از TF-IDF به تولید ویژگی می‌پردازیم:

```
def ngrams(n_g):
    # Generate n-grams
    cv = CountVectorizer(token_pattern=r"(?u)\b\w+\b", stop_words=None, ngram_range=(n_g,n_g), analyzer='word')
    dt_mat = cv.fit_transform(data['text'])
    # feature Generation. TF-IDF
    tfidf_transformer = TfidfTransformer()
    tfidf_mat = tfidf_transformer.fit_transform(dt_mat)
    return tfidf_mat
```

در پایان نیز داده‌ها را به براساس Test و Train و براساس Label و Text به ۴ دسته تقسیم کرده‌ام و پس از آن عملTrain را انجام می‌دهیم. گزارش کاملی نیز در پایان هر مرحله از Accuracy، Precision، Recall برای هر دسته داده شده و در پایان Confusion Matrix نیز نمایش داده می‌شود:

```
def train_test():
    # assign Train and Test sets
    X_train, X_test, y_train, y_test = train_test_split(
        tfidf_mat, data['label'], test_size=0.2, random_state=1)
    # Train
    clf = MultinomialNB().fit(X_train, y_train)
    # Test
    predicted = clf.predict(X_test)
    print("accuracy :", metrics.accuracy_score(y_test, predicted))
    print("classification_report :\n", metrics.classification_report(y_test, predicted))
    print("confusion_matrix :\n", metrics.confusion_matrix(y_test, predicted,labels=["ham","spam"]))
```