TripAdvisor Rating Classification from Reviews using NLP

Processing the Data and Converting the Text into Vector using Gensim Word2Vec

```
nlp = spacy.load("en_core_web_lg")

def preprocess(text):
  doc = nlp(text)

  filtered_token = []
  for token in doc:
    if token.is_punct or token.is_stop:
      continue
    filtered_token.append(token.lemma_)

  return(' '.join(filtered_token))
```

```
preprocess('my trip was ruined because someone robbed me')
```

```
'trip ruin rob'
```

```
df['Preprocessed_Review'] = df['Review'].apply(lambda x: preprocess(x))
```

```
df['Preprocessed_Review'] = df['Preprocessed_Review'].apply(lambda x : x.strip())
```

```
import gensim.downloader as api
wv = api.load('glove-twitter-200')
```

```
[==================================================] 100.0% 758.5/758.5MB downloaded
```

```
▶ 1.2s
nlp = spacy.load("en_core_web_lg")

def preprocess(text):
  doc = nlp(text)

  filtered_token = []
  for token in doc:
    if token.is_punct or token.is_stop:
      continue
    filtered_token.append(token.lemma_)

  return(wv.get_mean_vector(filtered_token))
```

```
df['Gensnim_vector'] = df['Review'].apply(lambda x: preprocess(x))
```

Final Dataset

	Review	Rating	Preprocessed_Review	spacy_vectors	Gensnim_vector
0	nice hotel expensive parking got good deal sta...	4	nice hotel expensive parking get good deal sta...	[0.18028633, 1.0664271, -2.634105, -0.08633499...	[-0.014720408, -0.010043328, 0.0011006314, 0.0...
1	ok nothing special charge diamond member hilto...	2	ok special charge diamond member hilton decide...	[-0.10305005, 0.025253873, -1.6846557, 0.15054...	[-0.010541894, -0.00069436536, -0.0042530587, ...
2	nice rooms not 4* experience hotel monaco seat...	3	nice room 4 experience hotel monaco seattle go...	[0.070874386, 0.408771, -2.0989778, 0.3434877,...	[-0.017286377, -0.0051590456, -0.0013687223, -...
3	unique, great stay, wonderful time hotel monac...	5	unique great stay wonderful time hotel monaco ...	[-0.62546104, 0.31182808, -2.3708231, -1.31422...	[-0.023345836, -0.0135795465, -0.007060945, 0....

Balancing the Unbalanced Dataset

```
X = df['Gensnim_vector']
Y = df['Rating']
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42, stratify = Y)
```

```
X_train = np.stack(np.array(X_train))
X_test = np.stack(np.array(X_test))
```

```
from imblearn.over_sampling import RandomOverSampler

oversampler = RandomOverSampler(random_state=42)

X_oversampled, y_oversampled = oversampler.fit_resample(X_train, Y_train)

X_test_oversampled, y_test_oversampled = oversampler.fit_resample(X_test, Y_test)
```

Training the model using Ensemble Learning(Voting Classifier)

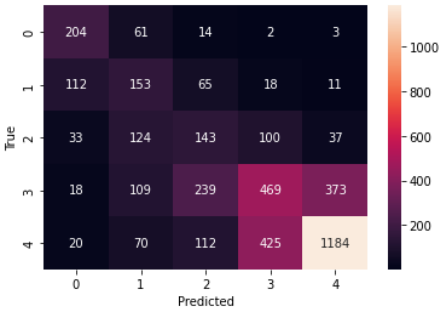
```
Pipeline
Pipeline(steps=[('Ensemble',
                 VotingClassifier(estimators=[('LogisticRegression',
                                               LogisticRegression()),
                                              ('RandomForest',
                                               RandomForestClassifier()),
                                              ('SVM', SVC()),
                                              ('KNN', KNeighborsClassifier()),
                                              ('AdaBoost',
                                               AdaBoostClassifier())]))])

Ensemble: VotingClassifier
VotingClassifier(estimators=[('LogisticRegression', LogisticRegression()),
                             ('RandomForest', RandomForestClassifier()),
                             ('SVM', SVC()), ('KNN', KNeighborsClassifier()),
                             ('AdaBoost', AdaBoostClassifier())])

LogisticRegression | RandomForest | SVM | KNN | AdaBoost
LogisticRegression() | RandomForestClassifier() | SVC() | KNeighborsClassifier() | AdaBoostClassifier()
```

Evaluation

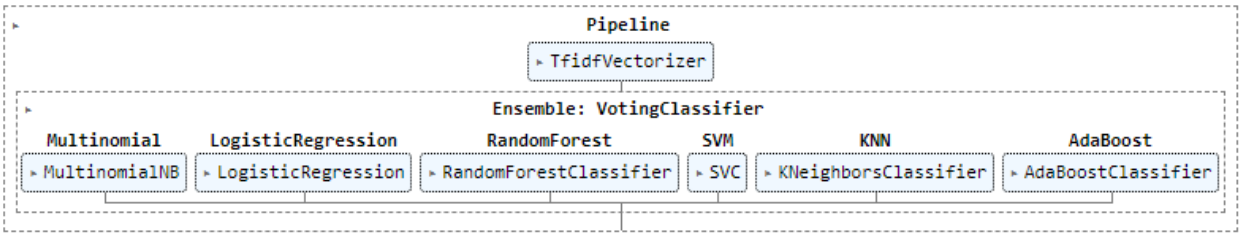
Classification Report

	precision	recall	f1-score	support
1	0.53	0.72	0.61	284
2	0.30	0.43	0.35	359
3	0.25	0.33	0.28	437
4	0.46	0.39	0.42	1208
5	0.74	0.65	0.69	1811
accuracy			0.53	4099
macro avg	0.45	0.50	0.47	4099
weighted avg	0.55	0.53	0.53	4099

Confusion Matrix



Using the TF-IDF Method



Evaluation(TF-IDF method)

	precision	recall	f1-score	support
1	0.65	0.70	0.67	284
2	0.40	0.46	0.42	359
3	0.36	0.32	0.34	437
4	0.52	0.52	0.52	1208
5	0.75	0.74	0.74	1811
accuracy			0.60	4099
macro avg	0.54	0.55	0.54	4099
weighted avg	0.60	0.60	0.60	4099